MAKE | BUILD | HACK | CREATE

132 PAGES OF MAKING

HackSpace

TECHNOLOGY IN YOUR HANDS

hsmag.CC | March 2020 | Issue #28

Mar 2020 Issue #28 £6

9 772515 514006 28

Raspberry Pi PRESS

ELECTRONICS

BUILD BETTER CIRCUITS

Strong 3D prints

Finding the settings to make plastic last

Raspberry Pi Audio

Make your projects sing

TESTING COMPONENTS
TOOTHPICKS
CIRCUIT PYTHON
BELT DRIVES

PROJECTOR MASK

Foil facial recognition with 3D printing and a Raspberry Pi Zero



SLA PRINTING ARCADES **STICK WELDING** VASE


8.9 MILLION+ PRODUCTS ONLINE | 800+ INDUSTRY-LEADING SUPPLIERS | 100% FRANCHISED DISTRIBUTOR
Your Proven Source
For the Components YOU Need
In Stock for Immediate Shipment
FREE SHIPPING
0800 587 0991
DIGIKEY.CO.UK
Digi-Key ELECTRONICS
*A shipping charge of £12.00 will be billed on all orders of less than £93.00. A shipping charge of $18.00 USD will be billed on all orders of less than $50.00 USD. All orders are shipped via UPS, Federal Express, or DHL for delivery within 1-3 days (dependent on final destination). No handling fees. All prices are in British pound sterling or United States dollar. Digi-Key is a franchised distributor for all supplier partners. New products added daily. Digi-Key and Digi-Key Electronics are registered trademarks of Digi-Key Electronics in the U.S. and other countries. © 2020 Digi-Key Electronics, 701 Brooks Ave. South, Thief River Falls, MN 56701, USA
ECIA MEMBER



Welcome to HackSpace magazine

Electronics is one of the keystone skills in making. Once you learn even a little bit, it opens up whole new possibilities. Pure electronics projects can be great fun, but so can adding lighting to woodworking, adding interactivity to clothing, adding automatic movement to 3D printing, and much more from an almost endless range of possibilities.

This month we're looking at levelling up electronics skills. If you're a complete beginner, **we'll walk you through your first circuit**

This month we're looking at levelling up electronics skills. If you're a complete beginner, we'll walk you through your first circuit. For more experienced makers, we'll look at adding microcontrollers, and other ways of adding more features to your circuits. Whatever level you're at, we'll help you build better circuits, and stay safe while doing it.

BEN EVERARD
Editor ben.everard@raspberrypi.org

PAGE 50
SUBSCRIBE TODAY



Got a comment, question, or thought about HackSpace magazine?

get in touch at hsmag.cc/hello

GET IN TOUCH

hackspace@raspberrypi.org
hackspacemag
hackspacemag

ONLINE

hsmag.cc

Available on the App Store

GET IT ON Google Play

EDITORIAL

Editor
Ben Everard
ben.everard@raspberrypi.org

Features Editor
Andrew Gregory
andrew.gregory@raspberrypi.org

Sub-Editors
David Higgs, Nicola King

DESIGN

Critical Media
criticalmedia.co.uk

Head of Design
Lee Allen

Designers
Sam Ribbits, Harriet Knight, Ty Logan

Photography
Brian O'Halloran, Abby Warren

CONTRIBUTORS

Lucy Rogers, Drew Fustini, Jo Hinchliffe, Sean Hodges, Mayank Sharma, Andrew Lewis, Glenn Horan, PJ Evans, Andrew Henderson, Gareth Branwyn, Les Pounder, Marc de Vinck

PUBLISHING

Publishing Director
Russell Barnes
russell@raspberrypi.org

Advertising
Charlie Milligan
charlotte.milligan@raspberrypi.org

DISTRIBUTION

Seymour Distribution Ltd
2 East Poultry Ave,
London EC1A 9PT
+44 (0)207 429 4000

SUBSCRIPTIONS

Unit 6, The Enterprise Centre,
Kelvin Lane, Manor Royal,
Crawley, West Sussex, RH10 9PE

To subscribe
01293 312189
hsmag.cc/subscribe

Subscription queries
hackspace@subscriptionhelpline.co.uk

recycle

This magazine is printed on paper sourced from sustainable forests. The printer operates an environmental management system which has been assessed as conforming to ISO 14001.

HackSpace magazine is published by Raspberry Pi (Trading) Ltd., Maurice Wilkes Building, St. John's Innovation Park, Cowley Road, Cambridge, CB4 0DS The publisher, editor, and contributors accept no responsibility in respect of any omissions or errors relating to goods, products or services referred to or advertised. Except where otherwise noted, content in this magazine is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported (CC BY-NC-SA 3.0). ISSN: 2515-5148.

Contents



120

06 SPARK

06 **Top Projects**
Creativity is all around us!

16 **Objet 3d'art**
Form meets function in 3D-printed steel

18 **Meet the Maker:** Andrew Ziminski
What it's like to use tools from 2000 years ago

22 **Columns**
Why CircuitPython is the future of digital making

24 **Letters**
Continuing our endless love for free-form circuits

26 **Kickstarting**
Clothing to signal your group identity

28 **Hackspace** Maker Works
They make things in Michigan – lots of things!

33 LENS

34 **Electronics**
The components and modules to make your circuits soar

52 **How I Made** The Mask
Paranoid about state surveillance? Make one of these!

58 **In the Workshop** Ultrasonic pong
Play this classic game without getting your hands dirty

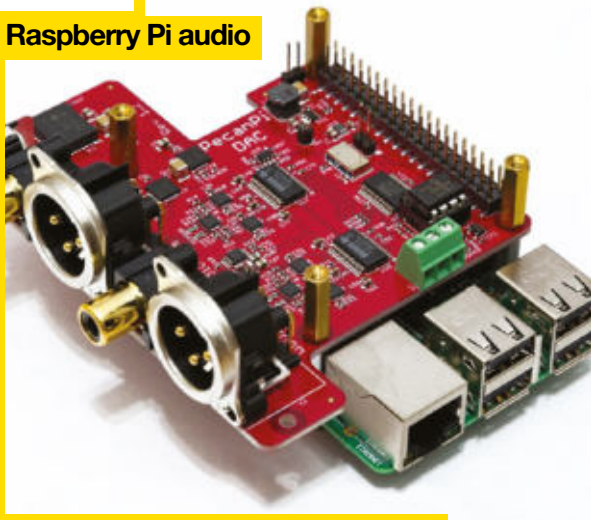
62 **Interview** York Robotics Lab
"Mummy, Daddy, where do robots come from?"

70 **Improviser's Toolbox** Toothpicks
Sharp mini tree-trunks for quick and easy builds

74 **Breaking 3D prints**
Test the strength of outlines and infill densities

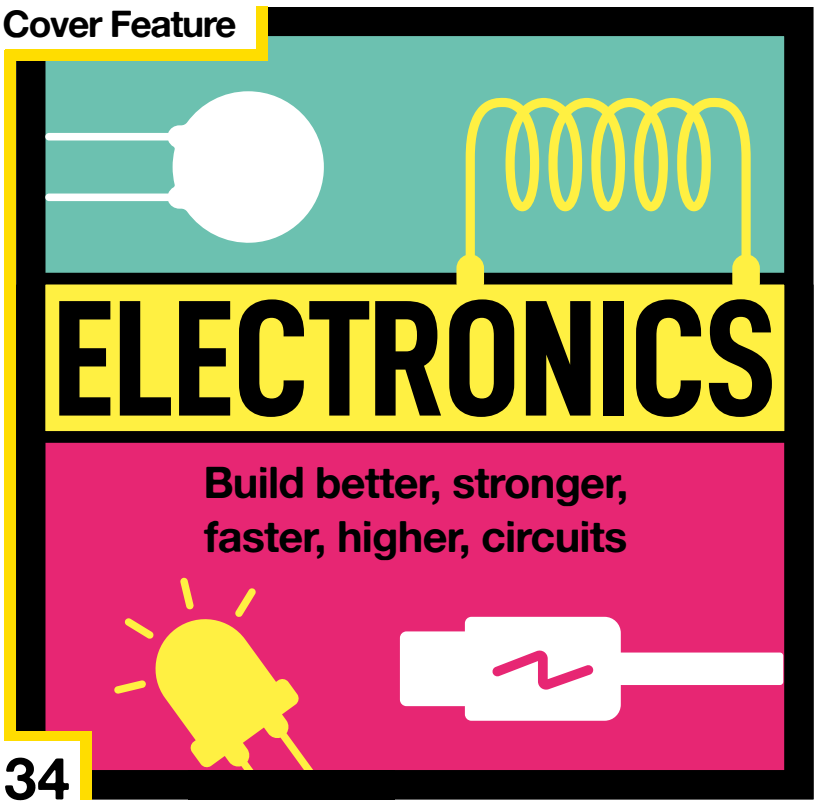
Tutorial

Raspberry Pi audio

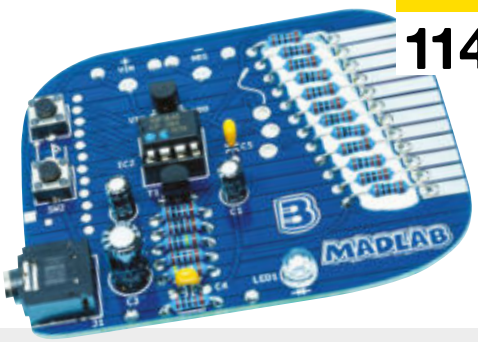


86 How to wrangle decent audio out of your Raspberry Pi

Cover Feature



34



114





16

96

Direct from Shenzhen

Component tester



112 Identify the odds and ends in your component drawer



06

Interview

Dr James Hilder



62 How does one go about building a robot? We asked an expert

77 FORGE

78 **SoM** CircuitPython
Manipulate the brightness of LEDs with dithering

80 **SoM** Precision boring
Make holes bigger with confidence and control

84 **Tutorial** Welders
Pick the right stick welder for your workshop

86 **Tutorial** Raspberry Pi audio
From beeps and fizzes to high fidelity sound

90 **Tutorial** Shop organisation
Attain tool-based omniscience

92 **Tutorial** Belt drives
Get power from where it is to where you need it

96 **Tutorial** 3D-printed vase
Explore effects in Cura

104 **Tutorial** BeagleBone gamepad
Low-latency IO using programmable real-time units

111 FIELD TEST

112 **Direct from Shenzhen** Component tester
Find out what spare bits and bobs you have lying about

114 **Best of Breed**
Electronics kits for young and old

120 **Can I Hack It?**
Add a coin slot to your builds and make a fortune

122 **Review** Prusa SL1 and CW1
Prusa's new resin printing system

126 **Review** Pokit
A handy crowdfunded pocket multimeter

128 **Review** AmbiMate MS4
Many sensors, one board, one vision

Some of the tools and techniques shown in HackSpace Magazine are dangerous unless used with skill, experience and appropriate personal protection equipment. While we attempt to guide the reader, ultimately you are responsible for your own safety and understanding the limits of yourself and your equipment. HackSpace Magazine is intended for an adult audience and some projects may be dangerous for children. Raspberry Pi (Trading) Ltd does not accept responsibility for any injuries, damage to equipment, or costs incurred from projects, tutorials or suggestions in HackSpace Magazine. Laws and regulations covering many of the topics in HackSpace Magazine are different between countries, and are always subject to change. You are responsible for understanding the requirements in your jurisdiction and ensuring that you comply with them. Some manufacturers place limits on the use of their hardware which some projects or suggestions in HackSpace Magazine may go beyond. It is your responsibility to understand the manufacturer's limits.

Steam-engine Nixie clock

By **Paul Parry** bad-dog-designs.co.uk

One of my proudest commissions was being asked to make a special steampunk clock for Theo Paphitis's 60th birthday.

The clock has two individual Nixie clock modules inside that are perfectly synchronised. This is done by sharing a 32kHz master clock signal between the two modules. The clock also generates smoke on cue – this is achieved by direct heating of a very light mineral oil with a timer that cuts out after a couple of minutes, otherwise you can't see the clock!

In addition, there is a miniature steam-engine as the centrepiece that is electrically driven for effect, with a speed control on the side.



Right
It's not just the electronics: we love the wood and brass on Paul's latest build


V+
130
300
600
LIMASSOL
To Theo on your 60th Birthday
From all of us at TPRC
Boux
AVENUE
Robert

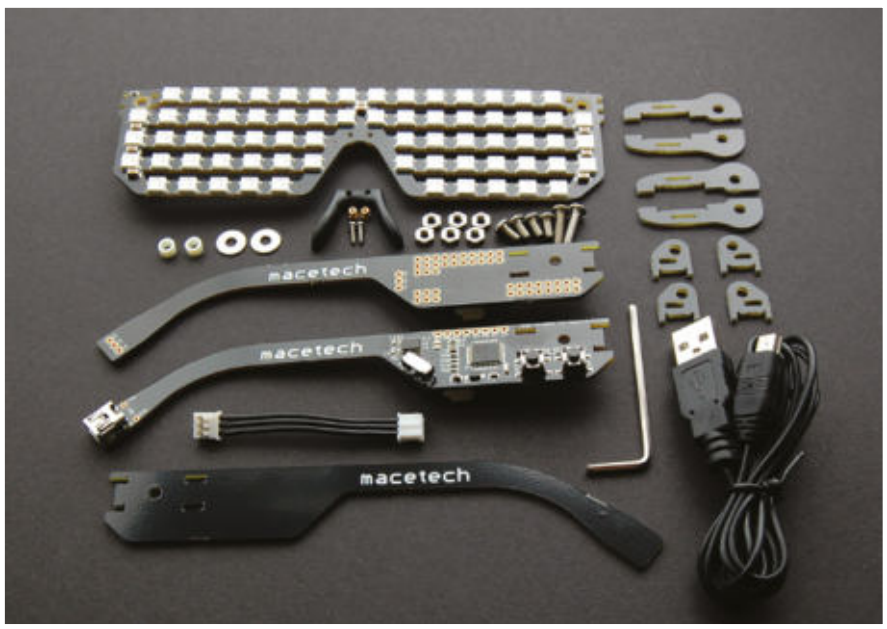


LED glasses

By macetech

hsmag.cc/p1JCIW

If you're a fan of addressable RGB LEDs, you've come to the right place. This kit from macetech, available on Tindie, uses an Atmel ATmega328 microcontroller chip to control 68 RGB LEDs. You can program it to show different patterns, colours, and even scrolling text – and, to keep construction simple, the front panel is a PCB.





Above
Put these on, hold a guitar, and you could almost be Muse's Matt Bellamy. Almost

Bourbon Barrel PC

By Josh Kroger

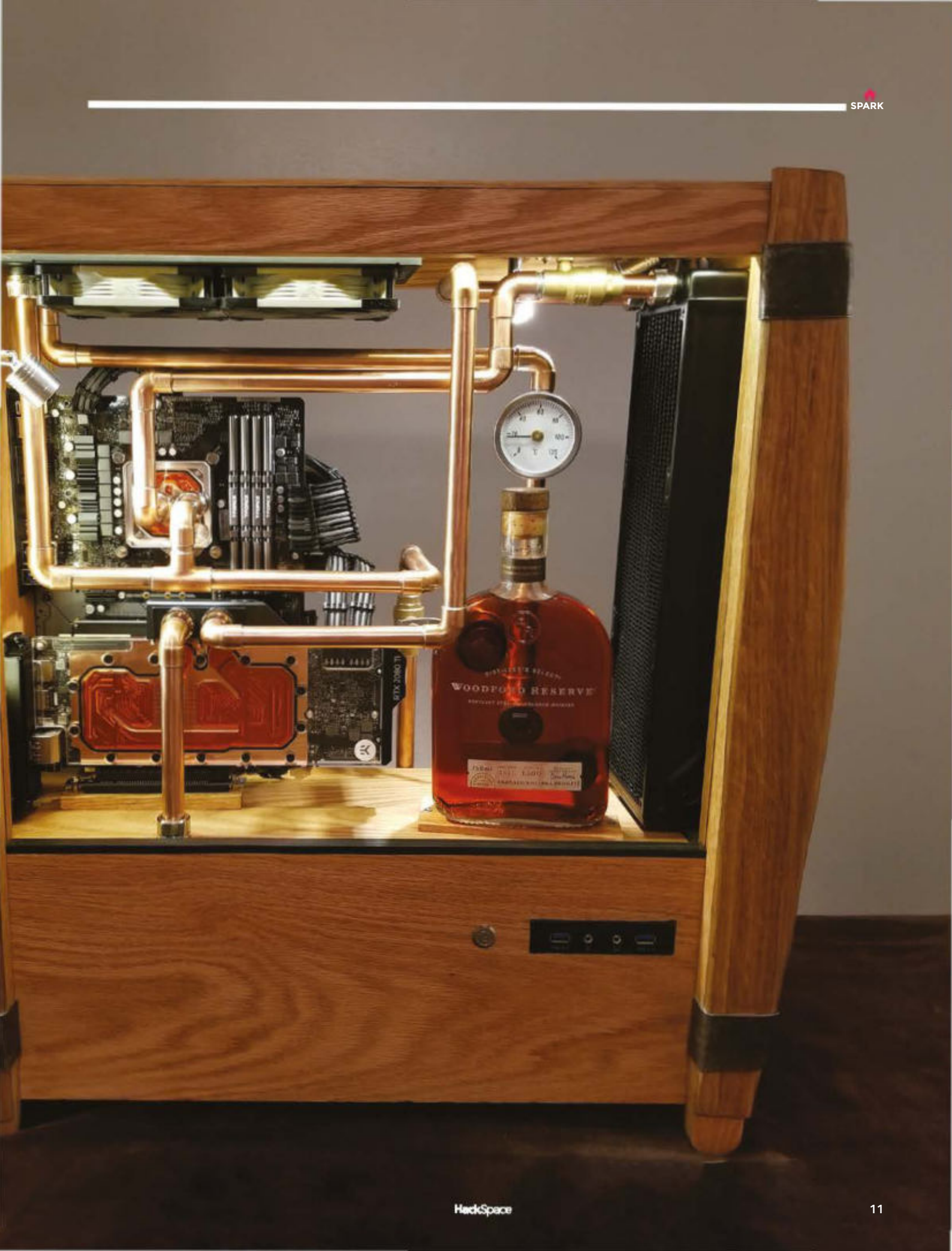
hsmag.cc/89f4IV

Whisky (or whiskey if you're into the stuff from Ireland or America) is lovely to drink, but it's even better as a coolant to take heat energy away from your overheating computer components. At least, that seems to be the thinking behind this absolutely perfect work from Josh Kroger, a 23-year-old mechanical and plumbing consulting engineer.

He's combined a bottle of booze and an old oak bourbon barrel to create a PC enclosure with a beautiful cooling system. It's fully functional, and the bottle really is plumbed into a working system.



Right
Remember, don't drink and drive – unless you're playing Mario Kart, that is


WOODFORD RESERVE

DNA sculpture

By **Paul Klinger** hsmag.cc/AHk4gV

You can make good money selling privacy, but you can make even more persuading people to give their data away. This 3D-printed DNA molecule by Paul Klinger gives away perhaps the most personal data of all, as the changing sequence of lights in the rungs of the DNA ladder gives away his own genetic pattern.

The A, C, T, and G bases are represented by green, blue, red, and yellow respectively – they're also displayed on a small OLED screen powered, like the lights, by a Raspberry Pi Zero.

Plastic parts are all printed on Paul's Prusa i3 MK3S, with the RGB LEDs in the spiral aligned with 4 mm PMMA fibre-optic cable, which is enclosed in opaque 3D-printed rungs that diffuse the light from the LEDs.



Right
Paul got his genetic information from dantelabs.com


Digital
BANANA FOR SCALE

REGULAR

LED LBD

By Debra Ansell

hsmag.cc/XhJXqs

Wearables are the easiest way to take your project with you, and a light-up dress is a bright way to spark interest from those nearby. This LED LBD (little black dress) is powered by an Adafruit Circuit Playground Bluefruit, using the Bluefruit phone app as a remote control to switch between display patterns. The Arduino code even incorporates data from the CPB's microphone in a sound-reactive VU meter mode. A video showing all the display modes is at **hsmag.cc/XhJXqs**.

Dress creator, Debra Ansell (blog at **GeekMomProjects.com**) believes that LEDs improve everything, and has been integrating them into clothing and accessories for almost a decade. She's even started a line of micro:bit-based codeable bags and back-packs at **BrightWearables.com**, to let everyone display how bright they are without needing to speak a word.



Right
Fashion comes and goes; the fascination with blinkenlights remains



Objet 3d'art

3D-printed artwork to bring more beauty into your life



Urwahn Engineering is a German company that specialises in steel-framed bicycles. Schmolke Carbon is a German company that specialises in carbon-fibre bicycle components. Together they've come up with this bonkers, brilliant city bike.

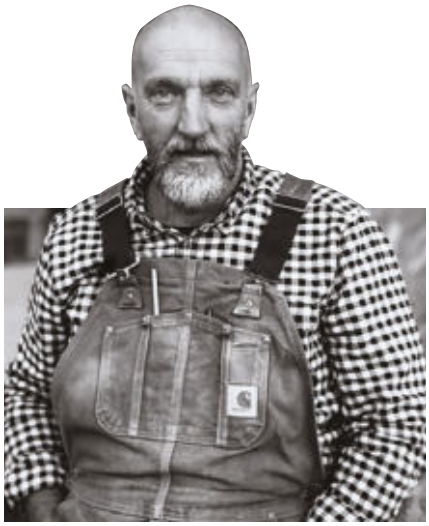
Urwahn has done away with the seatpost tube, integrating it into the rear forks to create a flexible back-end that takes inspiration from the hind legs of a fox.

Ramon Thomas, co-founder of Urwahn, says that 3D printing made sense from a functional as well as aesthetic perspective: "After we continued with precision casting, we realised that the walls where too thin for this production procedure," Thomas continues. "Therefore, we switched to selective laser melting, and managed to get very good surfaces and repeatable quality."

Selective laser melting is where instead of filament, the initial material is a powder bed (hence its other name of laser powder bed fusion). Lasers heat the powder at a given coordinate, and a solid metal form takes shape. Using this method, Urwahn has been able to integrate the rear forks with the seat tube, making the whole rear end of the frame flexible. □

urwahnbikes.com


5000
SCHWALBE
CONTINENTAL



Meet The Maker: Andrew Ziminski

Chiselling one rock at a time

It's a delightful fact that one of our oldest building materials, stone, is still being used today. It's good for the environment, looks great, and can be worked by a skilled hand into whatever shape you want it to be.

It's also a historical record: in most of Europe, you can look around and see the culture of the place written in the buildings.

To get an insight into the oldest making skill that's still in use today, we spoke to Andrew Ziminski. Andrew's a director of Minerva Conservation, and has worked across a huge number of buildings spanning the last 5500 years of human history.

"I've always been interested in the past. History was the only lesson at school that I was into. I wasn't at all academic, I just wanted to mess about.

> "It was fantastic – this was a medieval building that just hadn't been touched. **I learned more in that week than anything at school**"

"Everything changed when a museum was taking down one of the buildings in my town to rebuild it in an open-air space. I got involved with that when I was about 13, 14. It was fantastic – this was a medieval building that just hadn't been touched. I learned more in that week (that turned into a month, that turned into three months, that turned into a career) than anything at school. I learned about how the oak frame was held together, roofing material, and it had this big stone fireplace, a massive wall of stone.

"I was taking that apart, and I thought as soon as they put this back up I'm going to go back and volunteer, because I knew it was going to be a few years down the line. I left school and just bummed around for a bit, visited the museum, volunteered, and didn't look back. I helped rebuild this Elizabethan townhouse, and I did all the stone parts.

"There was an old guy there who knew everything, Mick. I stuck with him and learned a lot. I then went to Weymouth College, which has a well-regarded stone-masonry department, then I went to Salisbury Cathedral. I was fixing stones straight away right where the tower transitions into the famous spire.





Left
"No one's noticed, but the Gorgon has a little impression right in the middle of his nose. I think that's where they set out a massive pair of compasses to mark the central point, and they carved everything back from there"



"When we do Heritage Open Days, the number of people who want to come and have a go but who have never used their hands is amazing. You can see the mists parting for them. So many people have come back over the years to say they've signed up at college to do a stone-masonry course just as a hobby to keep their mind going. Working creatively in any way is better than bashing out an Excel file or whatever.

"There's a massive range of stonework out there. West Kennet Long Barrow is the oldest surviving structure in southern England, it dates to 3600-and-something BCE. And it's basically lumps of tabular rocks called Sarsens, which are completely undressed, but they've been laid out with a degree of mathematical consideration. We think it's the earliest use of geometry in a sacred space. It's an isosceles triangle inside. I've done work up there a few times, and I've got to know it quite well.

"That it uses the same big stones that are at Stonehenge, and the river Avon is not that far away; I think that these massive Sarsens were trimmed into a vague shape on the Marlborough Downs – they just lay on the surface of the chalk – and then they were rafted down the river to Stonehenge. Put out of your mind any visions of grunting savages dragging them for miles on logs and through forests; that's just nonsense. People have always moved heavy things by water. The stones for the pyramids were brought downstream on the River Nile. Portland stone was taken to London by boat. As a species, we've always moved heavy rocks by water.

"But the difference between West Kennet Long Barrow and Stonehenge is that Stonehenge was designed by an architect. The stones have been dressed and cut, and then formed into a scientific instrument to establish the position of the rising sun at midsummer and the setting sun at midwinter. Everything about Stonehenge is designed around those two dates.

"Stonehenge has mortise and tenon joints. It was designed by a carpenter. It's got tenons, which are the bits that stick up on the uprights. It's got lintels, horizontal stones that have these scoops taken out of them, and they have a tongue-and-groove effect to keep the rocks together. And it's completely over-engineered: these top stones weigh 30 tonnes, they're not going anywhere. It's very odd. →

Below
Andrew's book is on sale from 5 March. Buy it from a retailer that pays tax!





Above
It's important to discuss important tasks with key stakeholders

ANCIENT CRAFTSMANSHIP

"There would have been thousands of people employed in dressing, squaring off, erecting this thing, and then those people just disappeared. Why aren't there loads of other Stonehenges? You know how it goes: there are fashions, there's a new thing, then loads of other places pop up that emulate it. So that really puzzles me. Stonehenge came from nothing and disappeared into nothing. You go into the Iron Age, and there's nothing – stone wasn't used in any architectural sense until the Romans came. It's a bit of a puzzle. And then Roman architecture influenced Saxon architecture, which influenced Norman architecture, which became the Gothic. Islamic architecture gave us pointed arches and flying buttresses – basically Gothic architecture. There is a linear link, but it's very broken.



"The Roman stonemasons' toolkit is identical to the one I use today. I was at Caerleon Roman garrison museum this week doing a job in the village, and they've got a mason's toolkit that's exactly the same as mine, but 2000 years old. The hand chisels are a little bit squatter, but otherwise the same. The set-square and mallet are obviously the same – that's not surprising. But more than that, the methods of lifting

> Masonry naturally lends itself to carpentry repairs; **if you can do one, you can do the other**

that they used… we use these things called Lewis Pins, so you have a big block, you cut a hole in the top of the block and put these pins in, and they splay out in a certain way. They're exactly the same as we use today.

"I'm working on a medieval market preaching cross with a pyramidal roof on it at the moment. Masonry naturally lends itself to carpentry repairs; if you can do one, you can do the other. We've rebuilt this massive pinnacle, which is about ten metres tall, and applied plaster to repair the inside of the roof. That's one example, but we do everything: arches, flying buttresses, window tracery, every aspect of an old building really.

"We do lots of work on medieval bridges as well in the summer when the river levels are lower. That's always interesting because of all the wildlife. And there's graffiti everywhere; travelling by boat, rivers being the main communication point, you find ancient





graffiti right at the water level. A lot of it is to ward off ill spirits, to keep good luck on the bridge. You find ancient graffiti in a domestic setting too: daisy wheel marks, pentagrams, that sort of thing around fireplaces and doorways.

"We found a pentagram carved into a stone floor in front of a fireplace in a village nearby, which was a bit strange. There's one house I worked on on the edge of the Savernake Forest, near Avebury. It makes me feel a bit chilly thinking about it. It had medieval graffiti all over it from different periods. Everything we lifted up had something hidden under it. There was a dead cat under the hearth, a sheep skull up the chimney; there were tiny sea-shells buried under floor-boards. There were bottles and bones in the walls. It's as if all the generations who were living there felt as if they were under siege from evil spirits. I'm sure that's just how it was in those times: 'the milk's curdled, it must be the fault of the old witch down the road'.

"Someone said to me, 'You should write a book'. And my first reaction was that that's not my job. But why shouldn't it be my job? Working people don't write books, and that's partly why I wanted to do it.

"It's using the same intellectual skills as working a piece of stone, but it's less demanding on your body. But I'm more tired after a day writing than I am after a day working on the site. I appreciate the writers' craft more now. Anything that engages your brain to create takes energy.

"I also feel like migrants are getting a bum deal. I wanted to write a book to illustrate the role that migration has played in the development of our nation. I think it's something that's been largely overlooked. Migration of people and ideas. No Islam, no Gothic architecture, no Renaissance. It's something we've always had." □



What are the essentials?

Toolkit for the apocalypse



Lucy Rogers

@DrLucyRogers

Lucy is a maker, an engineer, and a problem-solver. She is adept at bringing ideas to life. She is one of the cheerleaders for the maker industry, and is Maker-in-Chief for the Guild of Makers: **guildofmakers.org**

Most apocalypses – events that result in huge destruction and change – give little or no warning.

The 1908 Tunguska event, when a large explosion occurred over Russia, flattened an estimated 80 million trees, over an area of 830 square miles. It was probably caused by an air-burst of a meteoroid.

The 1859 solar storm or Carrington Event, where an explosion (coronal mass ejection) from the Sun hit the Earth's magnetic field, caused auroras to be seen near the equator (rather than just the polar regions), and also caused havoc with telegraph systems.

If a Carrington Event-sized solar storm hit the Earth now, it would cause widespread disruption to our electricity supplies – and in turn, everything else that relies on electricity, such as communications, water and sewage pumps, and the internet!

I've also been reading and watching post-apocalyptic science fiction recently and have begun to wonder how I would react if a global disaster did happen. As I am a maker, my mind has turned to what tools I would want with me. Yes, this all depends on the type of disaster – something where everyone just 'disappears' but all the infrastructure remains is a different problem than if everything is now four feet under water. However, here's my generic list of things I'd try to carry with me:

- Fixed blade knife
- Notebook and pencil
- Marker pen (Sharpie)
- Multi-tool (Leatherman Wave)
- Work gloves
- Adjustable wrench
- Duct, insulating and self-amalgamating tape
- Torch (5 V rechargeable)
- Diamond file
- Paracord
- 5 V, high-current power bank / battery
- Solar panel (for charging power bank)
- Buck or step-down converter
- Kindle with a lot of reference books, and all the HackSpace magazines downloaded onto it
- 5 V soldering iron, lead free solder, solder wick
- Blu Tack
- Breadboard jumper wires and crocodile clip cables
- Spudger – plastic and metal to open up electronics and salvage the parts
- Craft knife
- Hot-melt glue stick (not the gun)
- Silicon tube (for syphoning water etc.)
- Pneumatic tube
- Solid core and stranded wire
- Magnifier
- Side cutters
- Needle-nose pliers
- Pocket multimeter
- Small, strong magnet
- Allen key set
- Socket set and screwdriver heads

A comprehensive first aid kit would also be a must. I wonder what I'll be kicking myself for having forgotten!

With thanks to Brian Corteil (**@CannonFodder**) and Richard Sewell (**@Jarkman**) for their toolkit shares, which gave me inspiration.

Python on hardware

The future is serpentine



Drew Fustini

@pdp7

Drew Fustini is a hardware designer and embedded Linux developer. He is the Vice President of the Open Source Hardware Association, and a board member of the BeagleBoard.org Foundation. Drew designs circuit boards for OSH Park, a PCB manufacturing service, and maintains the Adafruit BeagleBone Python library.

Python is an extremely popular programming language used for all sorts of things, from web and app development, to programming single-board computers and microcontrollers. Python is what most people use to program their Raspberry Pi, MicroPython is the text-based language of choice for the micro:bit, and Adafruit is investing heavily in CircuitPython, which is now available on many of their boards. Until recently, most maker-oriented microcontrollers, including Arduino, used languages based on C and C++. So why is this changing? And what difference does it actually make?

Let's start with the fundamental differences. C and C++ are compiled languages. When you upload code to an Arduino, your computer compiles your code into binary instructions, and then sends those instructions to your Arduino. Python is an interpreted language. When you save MicroPython code to a micro:bit, it does not get translated into binary by your computer. Instead, we install MicroPython on the micro:bit, which is then able to interpret and execute the code you send it without compiling and reflashing.

This means that Python-based boards have a much quicker prototyping cycle: as soon as you press save, your code will start running. Python is also more readable, making it seem less daunting to people who aren't already programmers. For organisations such as micro:bit and Adafruit, this is key: making hardware welcoming to beginners will increase the size of their community.

There are two main trade-offs: speed and availability of libraries. Compiled languages are faster, and there are currently far more libraries available for Arduino C++ than any other hobbyist language.

The two most popular implementations of Python for microcontrollers are MicroPython – created by Damien George for STM32, and now used by micro:bit and ESP32 – and CircuitPython – a fork of MicroPython that Adafruit manages, focussed on the Atmel ARM SAMD family and a Nordic BLE microcontroller. They are similar, but CircuitPython requires the microcontroller to support USB, and has sacrificed some functionality in favour of simplicity.

> **Another exciting development is that Python can also be used to actually design hardware**

Another exciting development is that Python can also be used to actually design hardware, thanks to open-source projects such as Migen and LiteX. Migen is a Python framework for describing digital circuitry at a higher level than Verilog. LiteX builds upon Migen to enable you to build a fully functioning system on a chip (SoC), including a processor core, that can be loaded into an FPGA. Leveraging the accessibility of Python for chip design is a fascinating idea – I can't wait to see how it evolves!

Letters

ATTENTION ALL MAKERS!

If you have something you'd like to get off your chest (or even throw a word of praise in our direction) let us know at **hsmag.cc/hello**

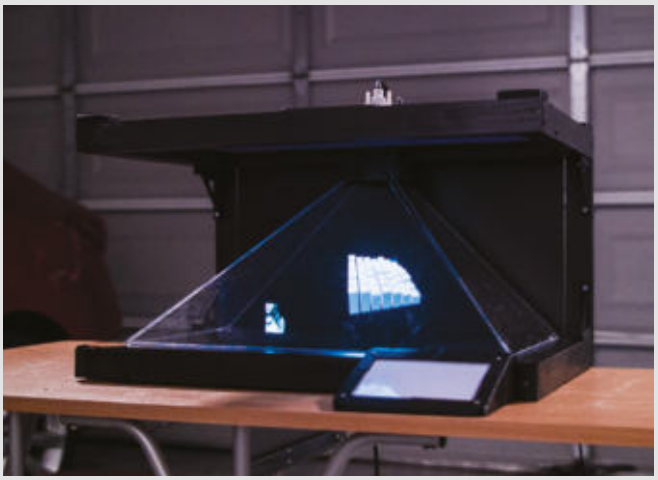
THE THIRD DIMENSION

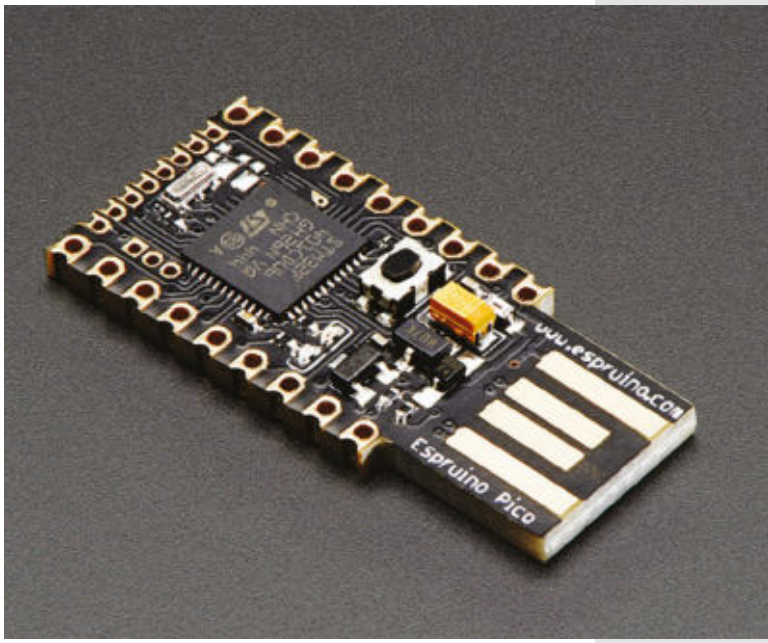
I can't believe that the holographic audio visualiser is a real thing! I'm sure I've seen something similar in a sci-fi film somewhere. The fact that it's made using just a regular monitor and a bit of acrylic is mind-boggling. How did I not know this was possible before?

I'm off to build holographic interfaces for all my household gadgets. BRB.

Leigh
Stevenage

Ben says: Yes, these holographic displays are both visually stunning and incredibly simple. You can get miniature versions of the acrylic pyramid for just a few pounds on many online stores. They're designed to turn mobile phones into holographic displays, but there's no reason they wouldn't work on any miniature screen whether it's attached to a Raspberry Pi, Arduino, or anything else. Movement is the key to getting a good visual impact.





ERRORS IN THE SCRIPT

In issue 27 on page 117, JavaScript and Java are confused. They aren't the same thing.

Joseph

Ben says: We listed the Espruino Pico – a microcontroller that runs JavaScript – as 'A good choice for Java developers'. This isn't necessarily wrong, but the two languages do indeed have little in common, and the Espruino Pico is an even better choice for JavaScript developers.

A WORK OF GENIUS

Wow, just wow! That's the only word I have for Jiří Praus's work. I don't know which is more impressive, the tulip, the heart, or the Arduino – they're all my favourite.

LEDs and brass seem to go so well together. There's something strangely futuristic about the harsh lighting combination. I wonder how far this can go? Could I build a computer out of just components and brass? How about a hi-fi? I guess the best way is to pick up my soldering iron and start practising.

Alex
London

Ben says: Yes, they are things of beauty. There's no real limit to the complexity of things you can build with brass, but if you're trying to do anything with high-speed signals (such as the ones between a processor and memory), you might find that strange things start happening. There's a whole host of design considerations for these fast signals that need to be taken into account. A hi-fi though, absolutely. That would be an ace project. Let us know how you get on.

HONOUR AMONG THIEVES

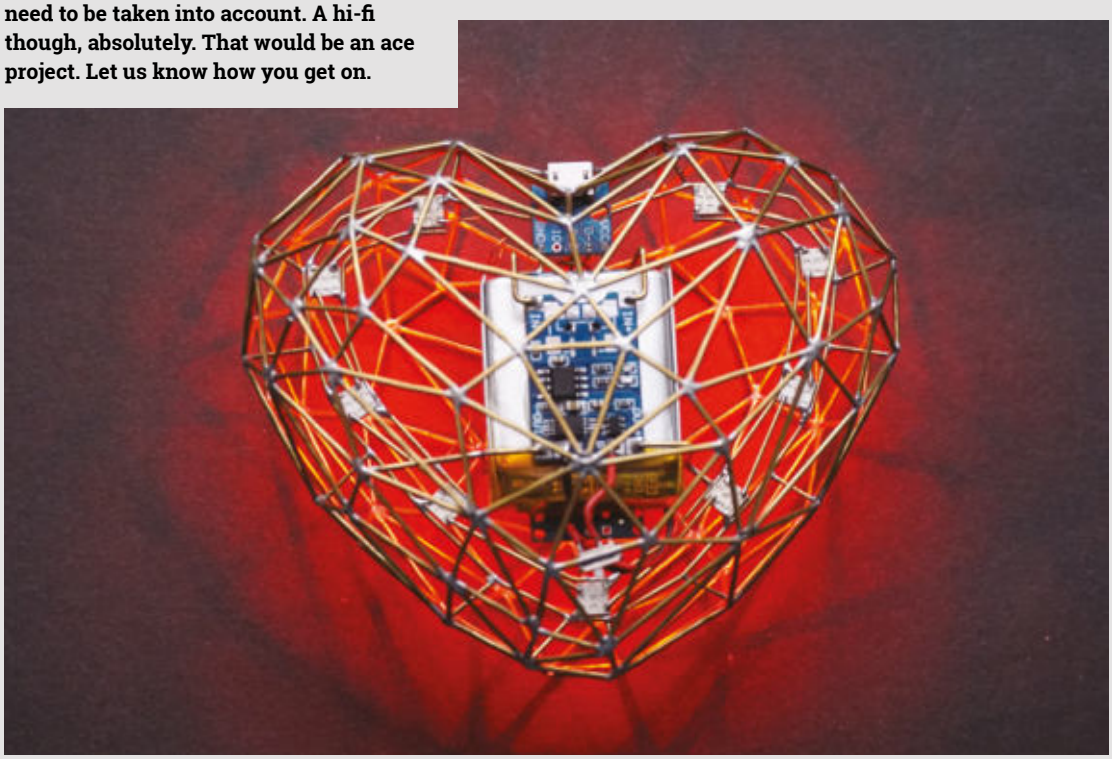


I'm a software developer by day, but I really enjoy making. I'm not very confident at all with electronics. I've been using HATs on my Raspberry Pis and shields on my Arduino, but it's a complete mystery to me how the good folks at Pimoroni, Adafruit, and others design these. I get the basics, but how do resistors, capacitors, and transistors power LCD screens or drive motors?

I was interested to read Simon Monk's tutorial on building a Joule Thief in issue 27. I think I just about understood what was going on, and it felt good to start with a box of components rather than pre-built modules. It's good to push myself as a maker and start building circuits. Can I have some more, please?

Ben
Birmingham

Ben says: I've got some good news for you about this issue! Also, keep an eye out in future issues for more electronics projects, it's something we're keen to feature more on. If there are any particular things you'd like us to cover, get in touch. Awesome name by the way!



REGULAR

CROWDFUNDING NOW

Mosfet girl

The people behind the silicon

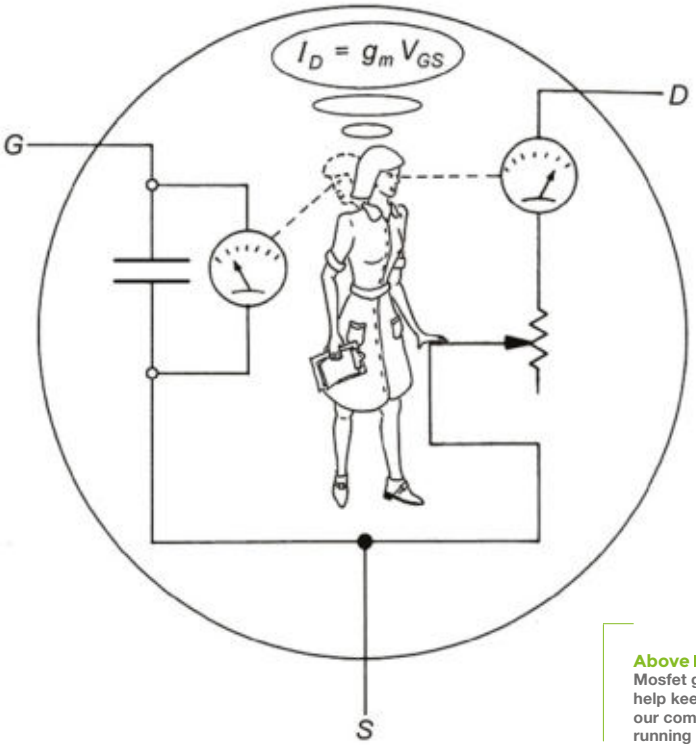
From $25 | **crowdsupply.com** | **Delivery: March 2020**

It's a little-known fact that inside every metal-oxide semiconductor field-effect transistor (MOSFET), there's a tiny woman. She monitors the gate, and adjusts a teeny-tiny variable resistor to make sure the current through the drain is appropriate. She's known as Mosfet girl, and it's thanks to her (and billions of her sisters) that we have modern electronics. She's the counterpart to Transistor man, who does a very similar job in bipolar junction transistors (BJTs).

OK, this isn't completely true. It's an analogy from the popular Horowitz and Hill book, *The Art of Electronics,* that has helped a generation learn the ins and outs of getting electrons to do their bidding.

This T-shirt is designed by Open Music Labs, but all profits from the crowdfunder will go to support ChickTech, a non-profit dedicated to increasing the number of women and girls pursuing STEM-based careers, and retaining women in the technology workforce.

This T-shirt looks great by itself, but the fact that supporting the crowdfunder is also helping increase diversity in tech is a double win.



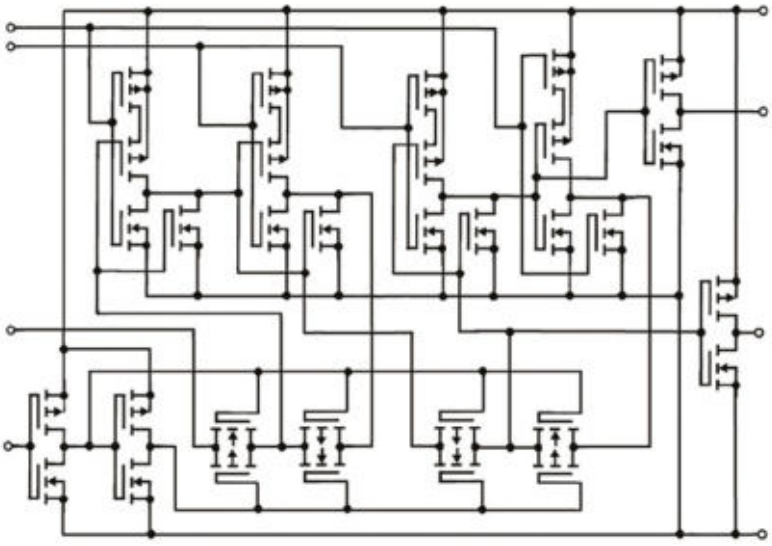
Above Mosfet girls help keep our computers running

BUYER BEWARE

When backing a crowdfunding campaign, you are not purchasing a finished product, but supporting a project working on something new. There is a very real chance that the product will never ship and you'll lose your money. It's a great way to support projects you like and get some cheap hardware in the process, but if you use it purely as a chance to snag cheap stuff, you may find that you get burned.







> **This T-shirt is designed by Open Music Labs,** but all profits from the crowdfunder will go to support ChickTech







Space of the month: Maker Works

Maker Works

maker-works.com

MakerWorx

makerworks_a2

Maker Works is a 1300 sq metre (14,400 sq ft) makerspace in Ann Arbor, Michigan, USA. Started in 2011, it offers a full machine shop with lathes, CNC and manual milling machines, MIG and TIG welding, iron worker, CNC plasma cutter, and sheet metal tools; a full wood shop with 4'×8' CNC router, table saw, band-saws, jointer, planer, and sanders; a jewellery studio with four-axis CNC milling machine, kiln, centrifugal caster, oxyacetylene torch, and vacuum chamber; craft area with laser cutters, CNC embroidery, industrial sewing machine, vacuum formers, and vinyl cutter; an electronics lab with CNC PCB engraver, test equipment, reflow oven; media blaster and power coating; 3D printing; and a computer lab.



Many tools require a checkout class, but we also offer skill development classes in welding, woodworking, CAD, CAM, Raspberry Pi, Arduino, and Inkscape.

While we strive to make each area reasonably complete, some of the most interesting work by members spans several different areas – for example, custom sheet metal enclosures for electronics, thick slab tables with welded steel frames, 3D-printed models for jewellery casting, and so on. It is very stimulating to be around so many creative people, some of whom are nationally known for their skills.

We started off expecting our members to be primarily hobbyists, but it turns out that most of our members generate some income from their activity here. Area businesses get memberships for their employees to fabricate prototypes. Custom furniture makers combine wood and metal for striking tables. Startups will 'rent' an office as they develop their products, then move to larger space nearby in the office park, as they expand. We've had several successful kickstarters by members, one of which spun out an entire business.

Our mission is to empower people and businesses in technical, creative, and educational fields locally by providing a wide range of tools, a place to use them, and a community of makers that supports the acquisition and sharing of knowledge; and globally by promoting the establishment and sustainability of makerspaces everywhere, through the sharing of best practices.

Maker Works has a big emphasis on continuous improvement, and the use of recipes, or 'Standard Operating Procedures', to get our members safely up to speed, even if they've never used a piece of equipment before. Check out our website for a free 16-page pamphlet that we wrote about how SOPs can improve safety, and provide a better experience for members in a makerspace. From the start, we had a →


MAKER WORKS
presents
THE GREAT
Maker Race







constant stream of people interested in starting their own makerspace, and we're happy to give folks a behind the scenes tour to talk about operations. For the past few years, we've offered a week-long course where we cover a whole toolbox of operational tools for a safe, efficient makerspace. (Look for our Makerspace Operations Bootcamp.)

We're pleased to be part of a Michigan makerspace group that meets about once a month, rotating between makerspaces, and learning about how we each tackle the challenges of running a makerspace. We were at the 2019 Nation of Makers' NOMCON, spreading the word of using Lean principles in makerspaces.

Using Recipes for a Great Makerspace: **hsmag.cc/EVxjfW**.

Makerspace Operations Bootcamp: **hsmag.cc/5enETS**.



Above
A 1300 square metre maker space needs a hefty dust extractor…

Far Right
… especially when it's full of dedicated dust creators like this champion!





CONTACT US

We'd love you to get in touch to showcase your makerspace and the things you're making. Drop us a line on Twitter **@HackSpaceMag,** or email us at **hackspace@raspberrypi.org** with an outline of what makes your hackspace special, and we'll take it from there.



Build a Makerspace for Young People

Join our **free online training course on makerspace design** to get expert advice for setting up a makerspace in your school or community.

Sign up today: rpf.io/makerspace

Raspberry Pi

The Raspberry Pi Foundation. UK registered charity 1129409

#RPiLearn

HackSpace
TECHNOLOGY IN YOUR HANDS

LENS

HACK | MAKE | BUILD | CREATE

Uncover the technology that's powering the future

PG 52



HOW I MADE: THE MASK

Foil facial recognition with a Raspberry Pi Zero, a tiny projector, and a 3D-printed mask

PG 58

IN THE WORKSHOP: ULTRASONIC PONG

Play this arcade classic without having to share skin cells with the great unwashed

PG 62



INTERVIEW: YORK ROBOTICS

How to fill a robot with data without confusing its tiny little silicon brain

PG 34

ELECTRONICS

Build better circuits with our guide to electronic enlightenment

PG 70

IMPROVISER'S TOOLBOX: TOOTHPICKS

Alternative making suggestions for tiny wooden teeth-cleaners

PG 74

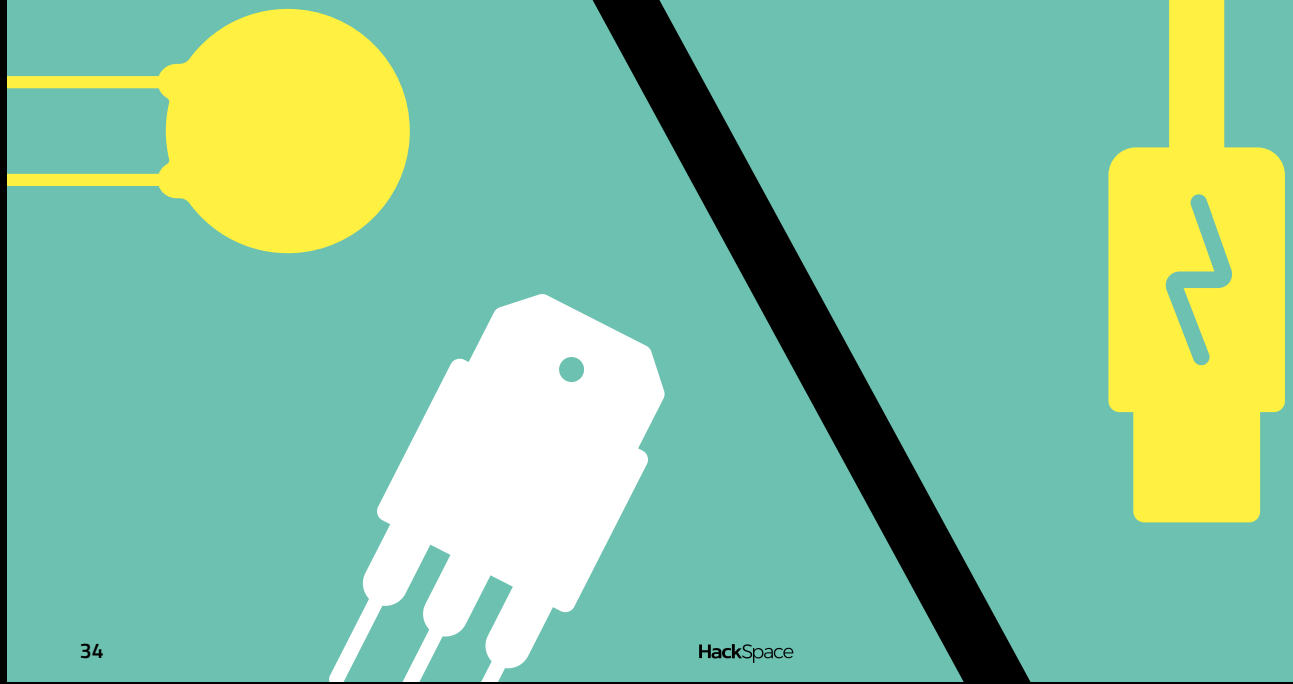
BREAKING 3D PRINTS

In the words of The Damned, everybody's smashing things now...

FEATURE

ELECTR

BUILD BETTER CIRCUITS



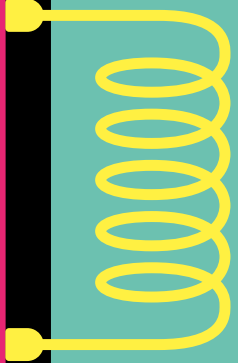
RONICS

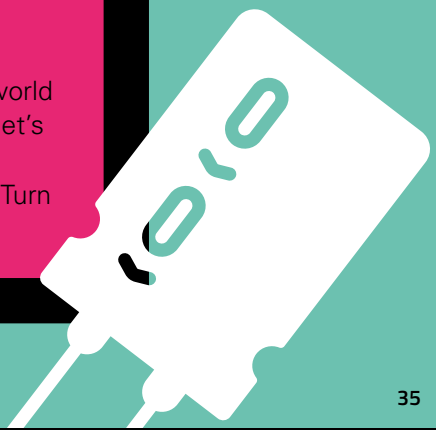


Electronics lets us add loads of interactive features to our projects. The word conjures up complex ideas of indecipherable circuit boards and strange components. However, it doesn't have to be complex. You can get a surprisingly long way with simple circuits with just a couple of components and, as you progress, you discover that each level of complexity is a simple step up from the previous one. As your skill and knowledge grow bit by bit, more and more features are available to you.

In this feature, we're looking at a range of skills in electronics, from the very basics for people new to the field, to ways of connecting up more advanced components, as well as some information for keeping yourself safe, which is important regardless of your skill level.

Building better circuits helps you make the world a brighter, more interactive, smarter place, so let's dive right in with the basics because, to build better circuits, you first need to build a circuit. Turn the page to get started. →





BUILDING YOUR FIRST CIRCUIT

TURNING ON AN LED WITH A FEW AA BATTERIES

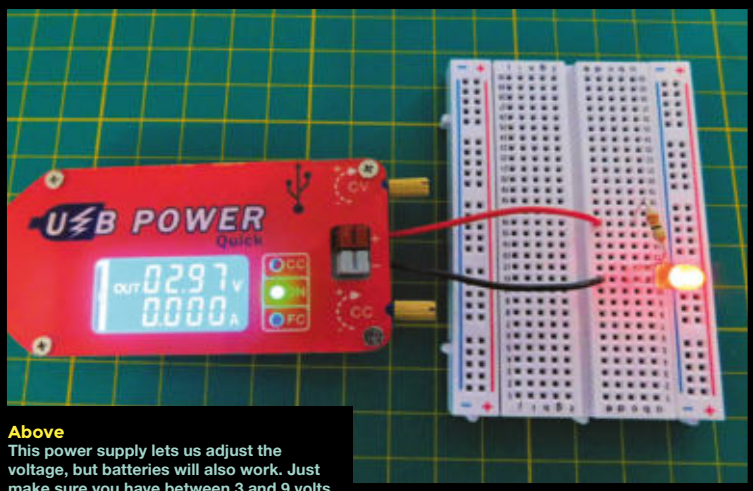
You've probably heard the term 'circuit' when describing something electrical. It might sound like technical jargon, but it describes the conditions needed for electricity to flow. A power supply, whether a battery or a mains-powered device, will only create an electrical current if there's a path all the way from the positive connection to the negative connection.

Let's start with a really simple circuit. You'll need a power supply of between three and nine volts – AA batteries are around 1.5V, so a two or four battery holder works well, as will a 9V battery. It could also be a bench power supply, but you're unlikely to have access to one of these if you're just getting started.

You'll also need a light-emitting diode (LED), a resistor, some wire, and a breadboard. A diode is a component that only allows electricity to flow through it in one direction. Light-emitting ones, as you may have guessed, emit light when electricity flows through them. If you're using less than four volts, you'll have better success with a red LED as (for reasons we'll discuss later) these need fewer volts to run.

One of the big challenges of building circuits is making sure that you don't provide too much electrical current or voltage to components because, if you do, they may go up in smoke. See the 'Current' and 'Voltage' boxes for more detail, but in simple terms, we need to make sure that the power supply isn't too big, and there's enough resistance to protect the components you're using.

We'll look at the exact calculations for resistance later, but for now, you just need to know that 330 ohms are fine for this circuit, but you'll learn how to calculate other options later.



Above
This power supply lets us adjust the voltage, but batteries will also work. Just make sure you have between 3 and 9 volts

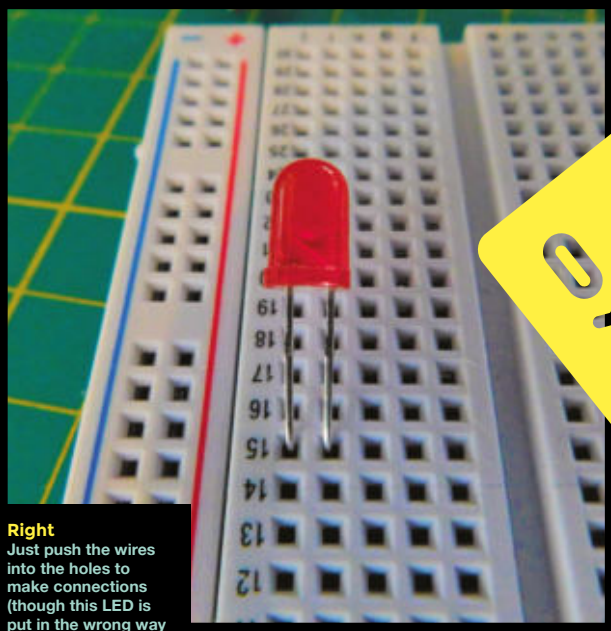
YOUR FIRST CIRCUIT

When you've got your components ready, you'll note that the LED isn't glowing. This isn't very surprising as there's no electricity flowing through it. In order to get it to glow, you need to create a complete circuit, and to do this, we'll use a breadboard. Breadboards are devices for prototyping circuits. They consist of two sets of parallel

VOLTAGE

As well as a current, a flow of electrons has a voltage. This is sometimes called a 'potential difference' or 'electromotive force' (EMF). Current will flow whenever two things of different potential differences are connected by a substance that conducts (typically a wire).

The relationship between potential difference and current is summed up in Ohm's law, which states voltage = current * resistance. We can restate this as current = voltage / resistance. This means that the amount of current (and hence the number of electrons) flowing through a circuit is equal to the voltage (potential difference) divided by the resistance, (which is the property for how much a particular material resists the flow of electrons). All conductors have some resistance, but for good conductors (such as wire), it's so low that we don't usually worry about it. Resistors are components that have a specific resistance, and we can use these when we want to limit the amount of current in a circuit.



Right
Just push the wires into the holes to make connections (though this LED is put in the wrong way around!)

rows, and each row is connected together. By putting multiple things in the same row, you can build a circuit.

It's good practice to build your circuit without the power supply attached. This means that there's no risk of accidentally connecting a circuit before the whole thing's ready. We'll start with our two components: the LED and the resistor. The LED is polarised – this means that you have to put it in the right way around. If you take a look at it, you should notice that one leg is longer than the other. The longer leg is the positive leg and has to connect to the positive side of the circuit. Put the LED in the breadboard with the longer leg connected to the resistor.

The next step is connecting up the power. First, connect the negative leg of the LED to the ground, earth, or negative side of the power supply. The final part of completing the circuit is to connect the resistor to the positive (or 5V) side of the power supply. At this point, the LED should start to glow – congratulations, you've made a circuit!

The important parts to remember are:

- Electricity needs a complete path around conductors in order to make a circuit (and therefore for current to flow)
- Too much current can damage components, so check if you need a resistor to protect them
- Too much voltage can damage components, so check the limits before connecting one up to a power supply →



Left
The humble resistor allows us to use delicate components without pushing too much current through the circuit

CURRENT

We don't want to get bogged down in the physics too much, but for our purposes, we can define electricity as a flow of electrons. Each electron has a tiny negative charge, and as they go around the circuit, they interact with components in different ways. We measure this flow of electrons in amps, with one amp being 6.24 × 10^18 electrons a second flowing past a point. We call this flow of electrons 'current'.

At this point, we hit our slight historical quirk. Electrons have a negative charge and therefore flow from the negative side of a power supply to the positive side. When we talk about 'electrical current', however, we talk about something that flows from the positive side to the negative. We don't need to worry too much about this, just remember that current is the flow of electrons (measured in amps, which is just really a measure of the number of electrons moving).

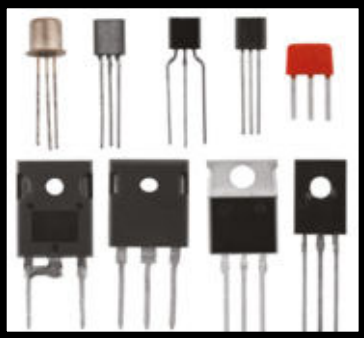
FEATURE

BASIC COMPONENTS

THE BUILDING BLOCKS OF ELECTRONIC CIRCUITS

While there's a mind-boggling range of components available from online electronics shops, they mostly fall into one of a small number of categories which we'll look at here. It's important that electronics enthusiasts can describe how they connect these different components together in circuits – this is done using schematic diagrams. In these diagrams, each component is shown using a symbol. Connecting wires are shown as lines connecting these different symbols.

We've taken a closer look at most of these components in earlier issues of HackSpace magazine. Where relevant, we've included a link to the free download of the appropriate issue.



Right
There's a vast range of transistor and transistor-like components available, and they all behave a little differently

TRANSISTORS

Transistors are perhaps the component that opens up the power of electronics more than any other. They allow one signal to switch another. Readers of a certain age may be most familiar with this from 'transistor radios' that used transistors to use the small current from the radio receiver circuit to switch the much higher current needed to drive speakers.

Modern computers are made of billions of tiny transistors, and hooking them up in different ways lets you build machines capable of complex computation. Their use is pivotal to modern computing – we looked at them in HackSpace magazine 13 (**hsmag.cc/issue13**).

Despite how essential they are to modern electronics, it's relatively rare to use transistors directly. Many makers and electronics hobbyists use them almost exclusively in integrated circuits which typically contain a lot of transistors, capacitors, and resistors all bundled together.

CAPACITORS ARE A CONCEPTUALLY SIMPLE ITEM



RESISTORS

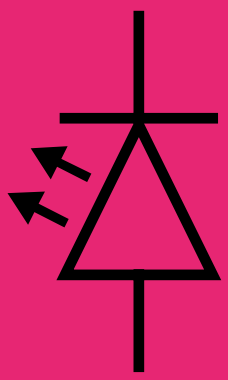


Of all the components, resistors are probably the simplest. They resist electrical current. The only thing you need to do is make sure that you're using the correct value resistor. This is displayed on the resistor using a series of coloured bands. By decoding these bands, you can know what the value is. It's worth getting used to this banding as it can make it quick and easy to find the ones you need, but while you're getting started, you can also use a multimeter to read the resistance.

We took a closer look at these in HackSpace magazine issue 9 (**hsmag.cc/issue9**).

Above
Resistors can be drawn either as boxes or zigzag lines – both symbols have the same meaning

LIGHT-EMITTING DIODE



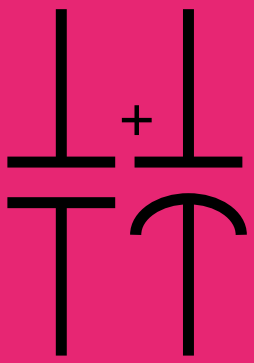
We saw on the previous page that LEDs emit light when electrical current flows through them. They come in several different colours, and there are even some multicolour types. They do have one slightly unusual electrical property. Because they're semiconductors, they don't have any resistance (OK, they have a very, very low resistance), but they do have a voltage drop. This is a bit confusing, but each LED has a forward voltage – from about 1.8 V for a red LED, up to 3 V or so for a white LED. There are two things you need this for. Firstly, if you supply an LED with less than the forward voltage, it won't light up. Secondly, when using Ohm's law to calculate the amount of resistance you need (see the previous page), you must take off the forward voltage from the total voltage. So, the current flowing through the circuit can be calculated with:

```
current = (total voltage - forward voltage) /
resistance
```

So, if we used a 5 V power supply, a 1.8 V LED, and a 220 ohm resistor, we would have 0.0145 A of current flowing through the circuit. We can also write this as 14.5 mA where mA stands for milliamps and 1000 mA = 1 A in the same way that 1000 millimetres are 1 metre.

Above
The triangle in the LED symbol points towards the negative leg

CAPACITORS



Capacitors are a conceptually simple item, but they spring up in a number of slightly unusual cases. They store small amounts of electrical charge. Perhaps their most common usage is for smoothing power supplies. A power supply won't produce an exactly constant voltage. There'll be spikes and dips as the circuit draws power. If you include a capacitor between the positive and negative connections on the power supply, it will store a small amount of charge (depending on the size of the capacity). If the power supply dips slightly, the capacitor will feed a bit of power into the circuit to smooth out the voltage.

This feature of storing charge also interacts in some useful ways with electrical signals that vary over time, such as audio signals, radio signals, and alternating current. To find out a bit more about how this works, take a look at the tutorial on capacitors from HackSpace magazine issue 10 (**hsmag.cc/issue10**).

Some capacitors are polarised, meaning that they have positive and negative legs, while others can be used either way around. Make sure you know which one you're using before putting it in a circuit.

Above
In some ways, capacitors can be thought of as very small rechargeable batteries

INTEGRATED CIRCUIT (IC)

ICs (or chips) are whole circuits made up of transistors, resistors, and capacitors bundled onto a single silicon 'die'. There's a variety of chips available that perform a wide range of functions. Some are simple (the 555 timer that featured in HackSpace magazine issue 19 – **hsmag.cc/issue19**), and some complex (the CPU in your computer).

All chips should come with a 'data sheet' that contains the information you need to connect it up. This should tell you what pin does what function, how much voltage it requires, etc. Another thing to look for is the physical format it comes in. Many older chips are available as dual in-line packages (DIP) that you can mount on a breadboard. Others come as surface-mount devices (SMD). These don't have 'legs' and can't be used with a breadboard, but have to be soldered directly to a circuit board. They may have little tabs for the solder, or they may have a grid of solder points on the bottom. →



Left
Because of the wide range of ICs available, there's not a set of symbols as such, but they're represented by boxes with text describing the model

FEATURE

MICROCONTROLLERS

GIVE YOUR CIRCUITS EXTRA POWER BY ADDING A PROGRAMMABLE BOARD

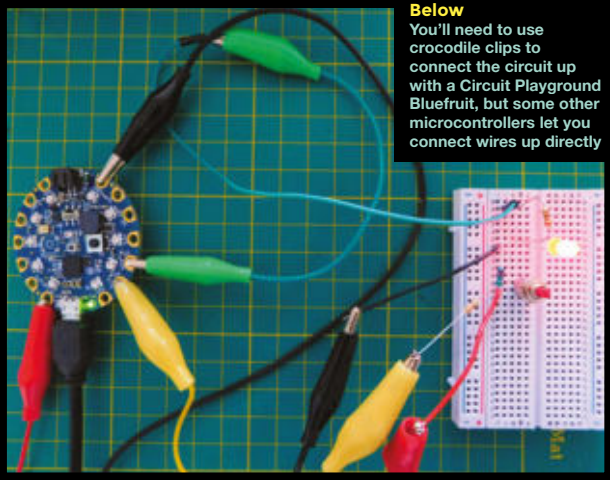
In the last exercise, we built our first circuit and lit an LED. It showed us the basics of circuits, but didn't really do anything useful. Let's take this knowledge and do something useful with it. In this section, we'll look at how to build circuits with a microcontroller. We used an Adafruit Bluefruit Express, but you could use any CircuitPython-compatible board. Using this and a few components, we'll build a reaction game where you have to press a button as soon as a light is turned on.

There are two parts to our circuit. The first part is much like the circuit we built to light an LED. The only difference is that rather than power our LED directly from a power supply, we'll power it from an input/output (IO) pin on the controller. These can switch on or off power on the pin from within the program. The voltage varies between devices, but most CircuitPython hardware runs at 3.3V. The main difference between an IO pin and a power supply is that we have to be careful about the amount of current we draw from an IO pin. While a battery may happily supply a few amps, IO pins are often rated for a few dozen milliamps. In the case of the Circuit Playground Bluefruit, you shouldn't draw more than 20mA from an IO pin. If we use a red LED with a forward voltage of 1.8V and a 330ohm resistor, the current should be:

```
current = (3.3-1.8)/330 = 0.0045 =4.5mA
```

That puts us well within our safe limit.

This circuit (see above) simply connects the A4 pin to the resistor, the resistor to the positive leg of the LED, and the negative leg of the LED to the ground. When power is switched on for the A4 IO pin, the LED will turn on, and when power is turned off, the LED will go off.

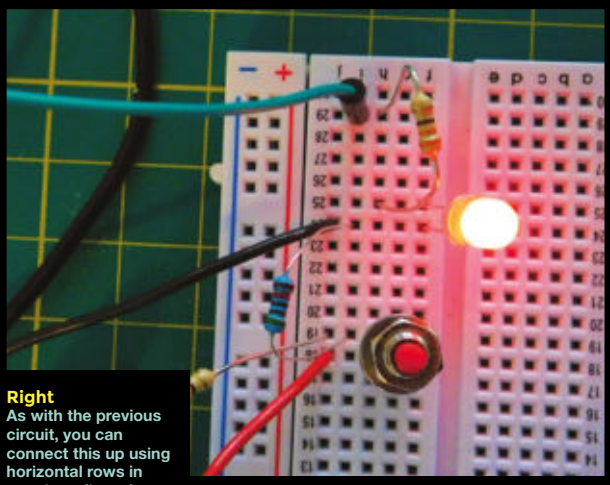


Below
You'll need to use crocodile clips to connect the circuit up with a Circuit Playground Bluefruit, but some other microcontrollers let you connect wires up directly

GETTING INPUT

We'll get input from a button, or more precisely, a momentary push switch. When these buttons are unpressed, there's no connection, so the circuit is broken. When the button is pressed, a connection is made, and current can flow through the component (provided it's part of a larger circuit).

We can start by attaching a button to the 3.3V output from the microcontroller. However, we can't connect the other side of this to the microcontroller because we have to be careful with current. As well as only being able to draw 20mA from the IO pin, we can only push the same



Right
As with the previous circuit, you can connect this up using horizontal rows in your breadboard

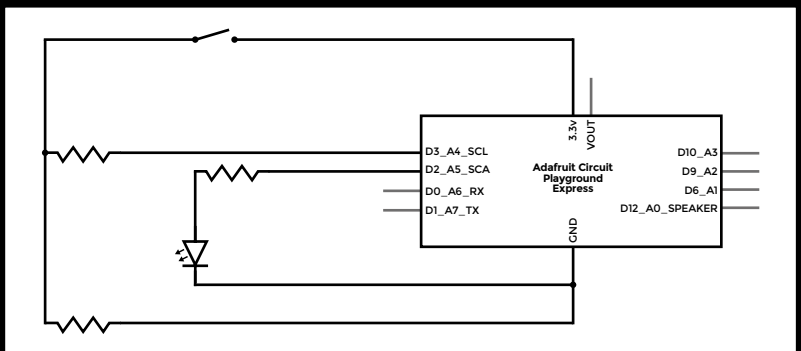
amount in. If we connected the button straight up, we'd have very little resistance and would damage the pin. We need a resistor, and we picked 330 ohms since it gives us a safe amount of leeway.

```
current = 3.3/330 = 0.01 = 10mA
```

Now, when the button is pressed, current flows into the IO pin, and this registers in the code as the pin having the logical value 'True'. When the button isn't pressed, the pin isn't connected to anything. This is known as 'floating', and since it's not part of a circuit, the CPU can't reliably read the state. The solution to this is to connect the pin to ground via a large resistor. We've used 10 kΩ, but similar sizes should work. When the button is pressed, most of the current will still flow through the button and into the IO pin (we'll lose a little bit through the larger resistor to ground, but not enough to cause any problems). When the button is released, there's still a connection to ground through the larger resistor, so this is registered as logical 'False'. The notion of True and False being 3.3 V and 0 V is an odd one, but it gives us a way of easily linking these notions between code and the circuit.

THE CODE

That's the circuit set up, let's take a look at the code now. We've used CircuitPython – if you have not used this language before, you can see how to get started with this link: **hsmag.cc/hDSwCi**.



Right
The circuit diagram of what we're building. It's useful to get used to building circuits from these diagrams as they're the standard way of describing circuits

The code defines two objects (**led** and **button**) that connect to the IO pins. The **led** IO pin is an output which means we can send values out to the pin while **button** is an input which means we read data from it.

In the main loop, we turn the LED off, then wait a random amount of time before turning it on. At this point, we check the time, then wait until the user presses the button. Finally, we display the amount of time the user took in the serial console.

```
import board
import digitalio
import time
import random

led = digitalio.DigitalInOut(board.A5)
button = digitalio.DigitalInOut(board.A4)

led.direction = digitalio.Direction.OUTPUT
button.direction = digitalio.Direction.INPUT

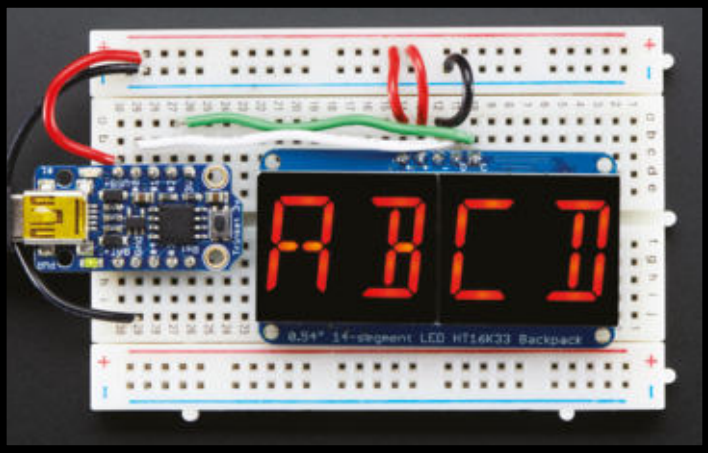
while True:
    led.value = False
    time.sleep(random.randint(5,10))
    led.value = True
    start_time = time.monotonic()
    while button.value == False:
        time.sleep(0.1)
    print(time.monotonic() - start_time)
```

This project is quite rudimentary, but shows how you can connect your circuits up to a microcontroller and in that way start combining code with electronics. This is a powerful combination, as it lets you use the processing power of modern computers to interact with the real world in an almost endless variety of ways. →

LOW-LEVEL PROTOCOLS

THE EASY WAY TO LINK UP COMPLEX COMPONENTS

Below
Here I²C is used to control an alphanumeric display using just two connections, data and clock



We've looked at how to connect up simple components such as LEDs and buttons, but what about more complicated things? Chips and modules that need more than just a simple on or off to work?

We need a way of sending, and there are a few standards that are widely used. Most of the time, when connecting up chips or modules you'll end up using Inter-Integrated Circuit (I²C), Serial Peripheral Interface (SPI), or Universal Asynchronous Receiver/Transmitter (UART). It's quite common for modules to come in different versions that work with different protocols, so you can pick the one that's right for you. You can think of these a little like the USB of the low-level electronics world. They are standards that tell you how to connect and send data.

Most hobbyist hardware platforms support all three, the only exception being smaller microcontrollers. Let's have a look at the differences.

CONNECTING THEM UP

UART is conceptually the simplest option. It connects two devices together, each has a transmit (Tx) pin and a receive (Rx) pin. Connect the Tx on one to the Rx of the other, and you're ready to go.

SPI is structured a little differently because it has a concept of a 'master' device that controls the communication with one more 'slave' device. There's a clock pin (often labelled SCK) that has to be connected between the master and the slave, then Master Out

Slave In (MOSI) and Master In Slave Out (MISO) also have to be connected between the master and the slave. You can also have more than one slave device, in which case the SCK, MOSI, and MISO are connected between the master and all the slaves, but there's also a 'slave select' (SS), sometimes called 'chip select' (CS). Each SS has to be connected to a different GPIO on the master device, and the master simply turns the SS on for the particular slave device that they want to communicate with.

I²C also allows multiple slave devices to connect to a single communications channel. There are two connections, Serial Data (SDA) and Serial Clock (SCL), and you simply connect together every data connection and every clock connection. Each device has an address that you can use to make sure you're sending data to the correct device. Addresses are hardwired into the devices, but you may be able to select between a few different addresses by connecting pins on the module to either GND or a positive voltage (VCC). This allows you to connect multiple identical modules to a single I²C bus without having address clashes. I²C connections also need two 2.7 kΩ resistors – one between the data channel and VCC, and one between the clock channel and VCC.

Regardless of the protocol used, as well as connecting the signal lines, you need to make sure the grounds are connected on each device. If you're powering them all off the same power supply, this will already be the case, but if they're powered separately, you'll need to add connections here.

All three protocols are widely supported by different programming environments, and there are libraries for Arduino, CircuitPython, and Python to implement each of them. Take a look at your environment's help for more information.

DANGER! HIGH VOLTAGE

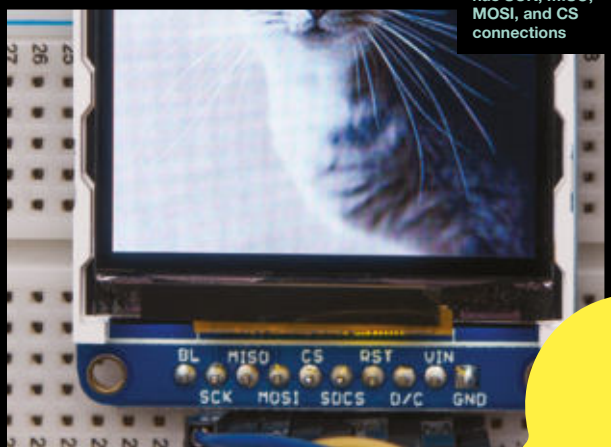
The protocols here don't specify a voltage, so can work with 3 V and 5 V circuitry. However, if you try and connect these two together, you might end up damaging your 3 V circuity. If you need to work with electronics using the different voltages, the easiest option is to use a logic level converter which you can simply put in the data and clock channels to switch the signals between the different voltage levels. For I²C, you'll need a bidirectional logic level converter, because the data line sends signals in both directions.

PULSE WIDTH MODULATION

Though perhaps not a protocol in the traditional sense, pulse width modulation (PWM) is a way of sending data via digital signals. PWM is about sending proportions (such as 0 to 100%) over a line that can only ever be on or off. It's based at a particular frequency (different controllers use different frequencies) – this frequency doesn't change, but the amount of time it spends turned on does. Take, for example, a PWM that runs on a ten-second cycle (real-world PWMs typically run hundreds of times a second). To encode 50%, you would have the signal on for five seconds, then off for five seconds. To encode 10%, it would be on for one second, then off for nine seconds. For 80% it would be on for eight seconds and off for two seconds. In each case, the total is ten seconds, but the proportion on or off determines the value.

WHICH TO CHOOSE?

UART, I²C, and SPI all provide ways of shuffling bits around, so you can use whichever one you want. However, there are a few limitations. UARTs only connect two devices together, and most controllers only have one or two UARTs, so if you need to connect multiple devices, UART isn't usually a great option (though you can get UART multiplexers if you need to do this). SPI's wiring can get a little out of hand if you've got a lot of devices, as each one needs an SS line and a corresponding pin on the controller. However, I²C needs each device to have a different address, and this is sometimes a problem (check the data sheet of your modules to see what the address is and if it's changeable). →



Below
You can see that this LCD module is SPI because it has SCK, MISO, MOSI, and CS connections

FEATURE

THE EASY WAY TO ADD FEATURES TO YOUR CIRCUITS

YOU DON'T HAVE TO BUILD EVERYTHING FROM SCRATCH



While you can build all your circuits from scratch (and it can be good fun to do this), if you're more interested in getting your project to work than detailed circuit design, there are loads of pre-built modules to help you. These are mini-circuits designed to provide a commonly needed feature. You can use them to provide some of the features you need, or even build complete systems by chaining them together.

Some are designed to work with microcontrollers or single-board computers and communicate with a protocol such as I²C or SPI, while others are more 'pure' electronics and can work on their own. We'll take a look at both types here to give you an idea of what's out there, and help you pick the best parts for your project.

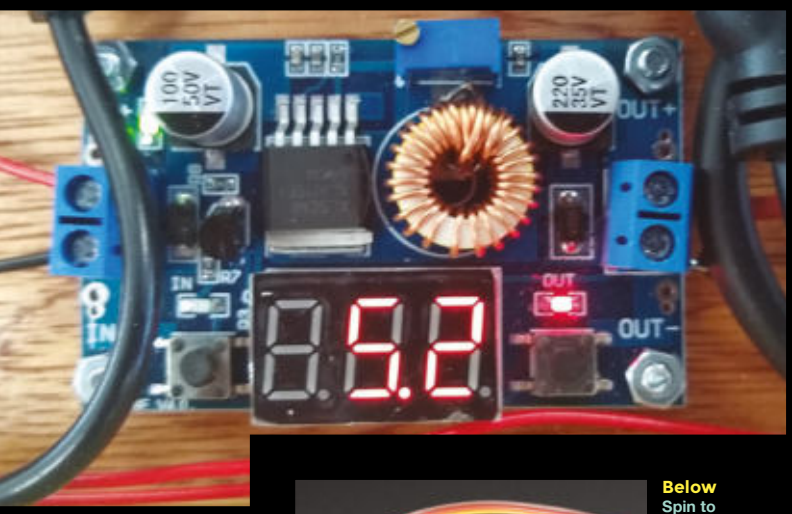
WHEN YOU PUT POWER INTO A MOTOR, IT SPINS

POWER

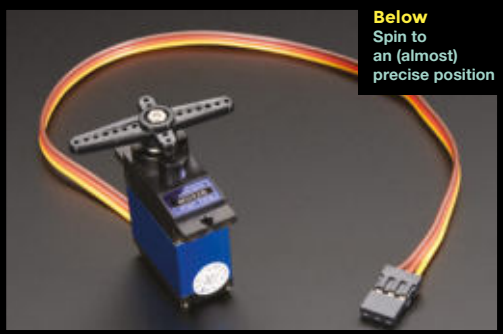
You'll often find that you have power coming into your project at one voltage, but need different voltages for different parts of your project. For this, you'll need a voltage converter.

Linear converters are the most simple. They take an input voltage and output a set lower voltage. The most common are 3V and 5V linear converters. They're simple to use as they only have three pins (voltage in, voltage out, and earth), but the downside is that the current going in is the same as the current going out. This means that less power goes out than comes in. Since power has to go somewhere, it's turned into heat, and they can get very hot.

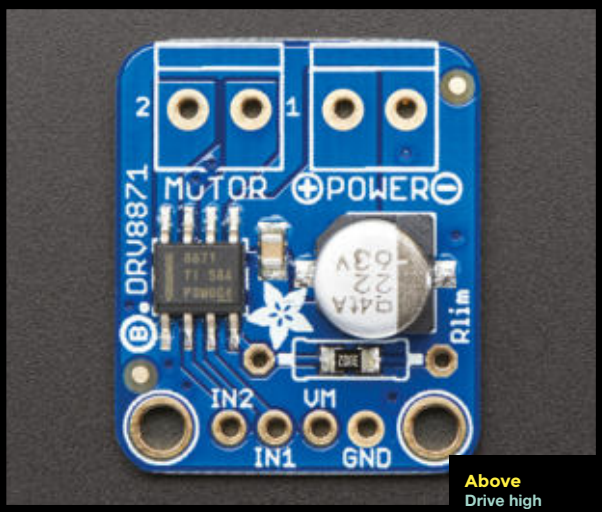
Switching converters are more complex, but can be far more efficient. They can be used to output a lower or higher voltage than comes in, but make sure you get the right type for the job. Buck converters output a lower voltage, while boost converters output a higher voltage. Both can be highly efficient, and you'll find versions that work at relatively high currents if you need that. One slight issue is that there is typically a little noise (random variation) in the voltage output. For many uses, this isn't a problem, but sensitive circuits may have problems.



Below
This switching regulator is powering our Raspberry Pi



Below
Spin to an (almost) precise position



Above
Drive high current loads from sensitive IO pins

MOTOR DRIVER

Most microcontrollers and single-board computers can only source or sink a certain amount of current from a GPIO pin. For low-power things like LEDs, this may well be enough, but for high-power things like motors, you'll need a bit of circuitry to get things running. You can build these circuits yourself, but the easiest option is a motor driver – these can drive anything that requires a large amount of current (and will be rated for a certain power that must be more than the motor draws).

The most common form of motor driver is an 'H-bridge' that takes four inputs and can drive the motor forwards, backwards, coasting, and braking. These inputs can be from microcontroller pins or any other input (such as a radio controller output).

SERVO

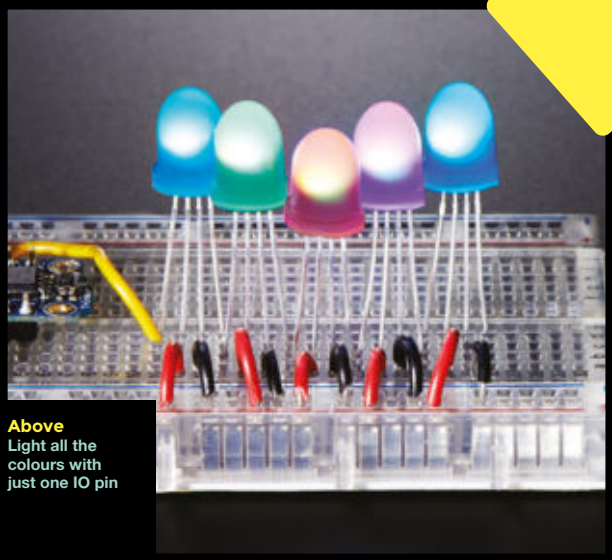
When you put power into a motor, it spins, but sometimes it's useful to rotate a motor to a particular position. There are a few ways to do this – the easiest is with a servo. These have three wires: power, ground, and signal. The signal line takes a PWM signal.

Depending on the signal in the PWM, the motor rotates to a different position. The results aren't brilliantly accurate, and if you need to go down to degree-level accuracy, you might want to consider a setup using stepper motors and gearing, but for many purposes, this isn't needed.

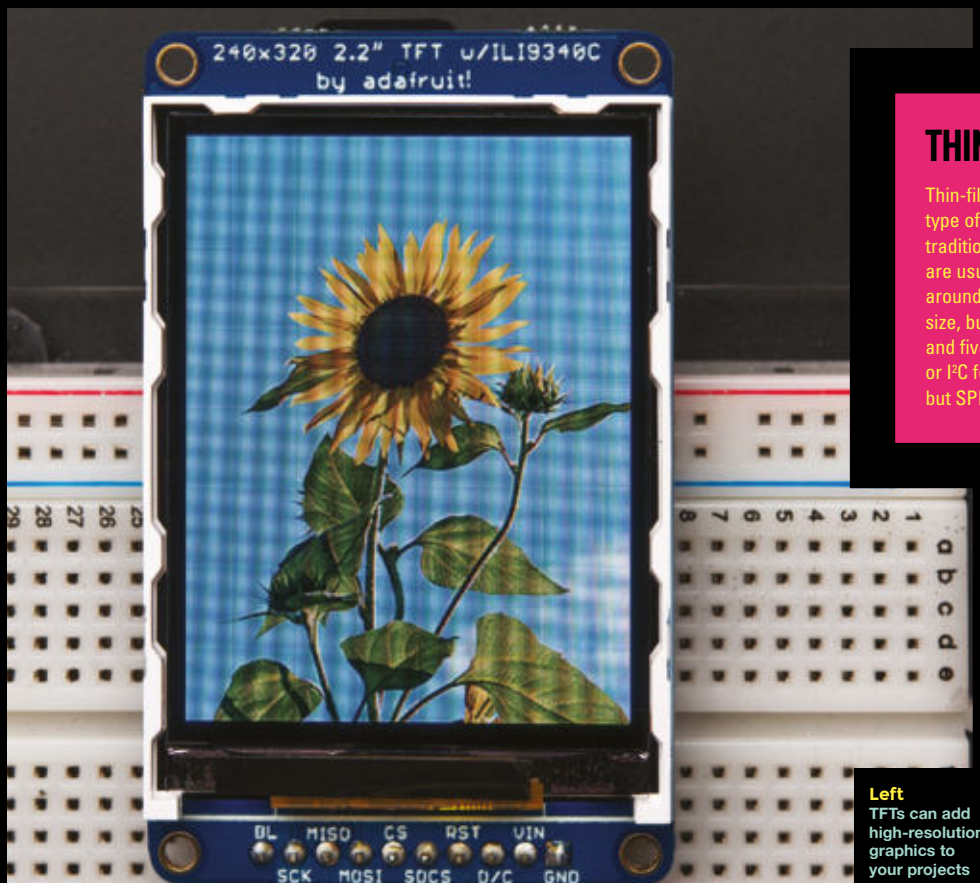
In most cases, you'll probably drive a servo from a microcontroller GPIO, but you can also get servo drivers that take their input from some other form (often a knob attached to a potentiometer).

COLOURFUL LEDS

It's really common to want to control LEDs that can show different colours. The basic device for these are LEDs that have red, green, and blue (RGB) LEDs all in one package. They have four pins, one for each colour and one common between them. You can hook these up as you would normal LEDs (the common connection is usually the negative connection, but not always). However, if you do this, you very quickly need lots of connections to your microcontroller. A much simpler option is chainable LEDs. These are LEDs with an integrated controller that understands some very simple communications protocol. The most common of these are NeoPixels (otherwise known as WS2812Bs). These have four connections, power, ground, data in, and data out. You just need to connect the data out of one to the data in of the next to create a long chain of RGB LEDs. The data in of the first can be connected to a GPIO pin on a microcontroller, and you can control them all with a single pin. There are libraries for all major microcontroller programming languages to help you control NeoPixels.



Above
Light all the colours with just one IO pin



Left
TFTs can add high-resolution graphics to your projects

THIN-FILM TRANSISTORS

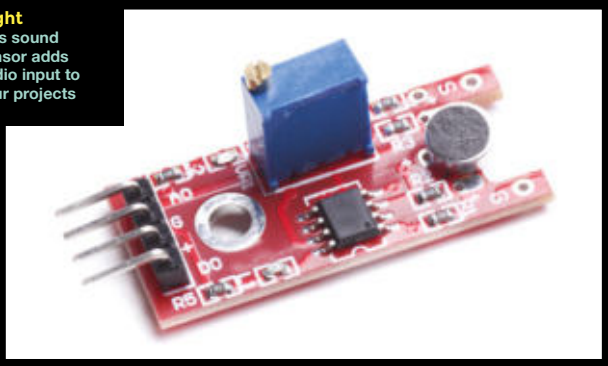
Thin-film transistors (TFT) displays are a special type of LCD that is a bit more responsive than traditional LCDs. They tend to be higher quality and are usually full colour. They vary in size from around one inch across, all the way up to monitor size, but typically maker models are between one and five inches in diameter. They often come in SPI or I²C forms. Both typically have the same features, but SPI can often achieve higher data rates.



ENVIRONMENTAL SENSING

There are a whole host of different environmental sensors, from temperature and air pressure (that are often in the same component) to gas sensors and myriad more esoteric sensors. Many of the more advanced sensors require a microcontroller to read the data over I²C or SPI, but some of the more simple sensors only affect the electrical properties. For example:

- Light dependant resistors (LDRs) change their resistance depending on the light falling on the surface
- Thermistors, similar to LDRs, vary their resistance, but this time in response to temperature
- Photodiodes work in a similar way to transistors, but light is used to trigger the flow of current, rather than electricity



Right
This sound sensor adds audio input to your projects

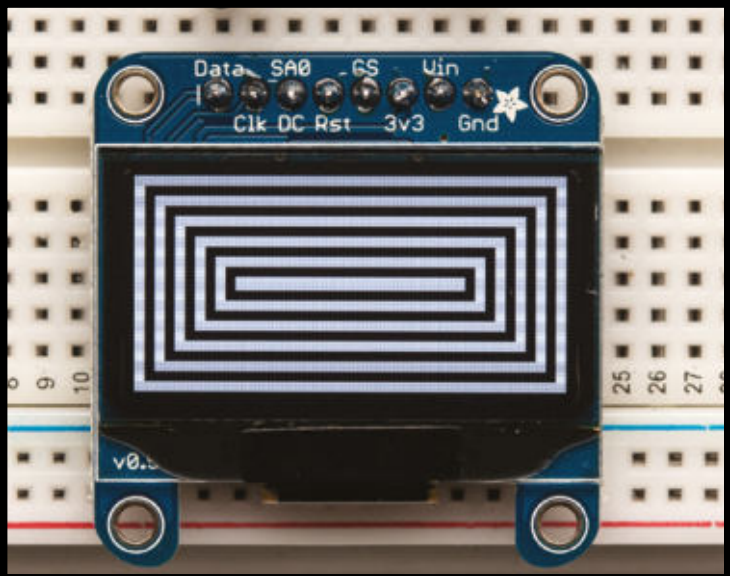


Above
If you need a little text, character LCDs are a great option

OLED

Organic light-emitting diode (OLED) displays are crisper and brighter than LCD and TFT displays, and can be a very affordable option, particularly the small, single colour options. Like TFT, the control instruction set can be a little complex, so it's best to check driver support for your microcontroller of choice before purchasing.

Below
Tiny, crisp OLEDs are cheap and easy to use



LIQUID CRYSTAL DISPLAY

Liquid crystal displays (LCDs) come in many different forms and at many different price points. Perhaps the most common for makers are designed to display lines of text, with each line being made up of a number of characters, and each character is a 5×7 grid of pixels that can represent the letter.

There are usually eleven data pins (plus a few for power and backlight), so they take a little bit of wiring, but once set up, they're easy to use with libraries for most common development environments. While it is conceivable you could drive these from non-programmable circuits, they're usually used in partnerships with microcontrollers.

SOLDERING SAFELY

KEEP YOURSELF SAFE WHEN MAKING ELECTRONICS



Above
The smoke may look pretty, but it can cause serious health problems

We ran an informal poll on Twitter to see how many people are using personal protective equipment when soldering. The results are in, and of the 643 people who answered, 62.1% used neither googles nor fume ventilation to protect themselves. 19.1% used only fume extraction, 9.3% used only goggles, and 9.5% used both fume extraction and goggles.

The biggest risk with soldering is fumes that can cause irritation of your throat and lungs, and in extreme cases, asthma. According to the British Health and Safety Executive (HSE): "Rosin-based solder flux fume is now regarded as one of the most significant causes of occupational asthma in Britain. When the asthmatic effects are fully developed, they are permanent and irreversible. Continued exposure, even to very small amounts of fume, may cause asthma attacks and the person affected may be unable to do any soldering with rosin-based fluxes again."

SAFETY

The information in this article is provided to help you consider your setup and encourage makers to think about safety. It is not definitive, and should not be considered complete safety advice. We'd encourage solderers to educate themselves further on the subject, and the HSE's leaflet '*Controlling health risks from rosin (colophony)-based solder flux fume*' (**hsmag.cc/W1Ewfq**) is an excellent source of information on the subject.

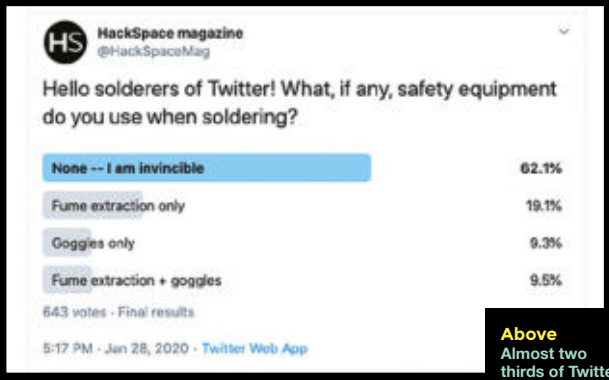
There are a few ways to protect yourself:

- The simplest way of reducing your exposure to soldering fumes is to think about posture. If you typically solder on a table, you may find that your head is directly over the item you're soldering. Raising the item so that it's at chest level or higher, and you're looking at it more from the side will mean fewer fumes breathed in. (The HSE do not consider this adequate for rosin-fluxed soldering, but if you're not prepared to do anything else, it will help.)

- Fume dispersers involve fans that suck fumes away from the person soldering. Some include a carbon filter, but testing by the HSE suggests that this filter does little to remove the dangerous particles. How effective these dispersers are depends a lot on where the fumes go. If you're working in an enclosed space, then they may not be enough to protect you from fumes. If it's sending fumes away from you and other people in the space, and towards a source of ventilation, then you'll probably have better success.

GOGGLES

Are goggles needed when soldering? It's never a bad idea to wear goggles when handling potentially dangerous liquids, and molten tin (with a few other metals mixed in) definitely qualifies as a dangerous liquid. While it's not particularly splattery, there's always a risk of bits of solder flying at your face. In our experience, this is more of a risk when desoldering, as the component could be under tension and flick once the solder is molten. Not many people solder with goggles, and very few get eye injuries, but the risk is there.



Above Almost two thirds of Twitter solderers don't use protection

BURNS

Soldering irons are hot! Burns are a risk, but gloves strong enough to protect you from burning yourself are usually incompatible with the amount of dexterity you need for placing components. Another burn risk is dripping solder – trousers and closed shoes are strongly advised!

- Fume capture systems can include a hood, an enclosed space, a soldering iron tip extractor, or a downdraught workbench. These can reduce fume exposure to zero or almost zero. They can be expensive to buy and need to be set up and used properly to be effective, but if you're soldering regularly, it's something that you might want to consider.

What the most appropriate method for protecting yourself is will depend on your particular circumstances, and how much you solder. Someone who only solders a few bits together a couple of times a year is in a very different risk category to someone who solders most days. It's up to each of us to make the decision for ourselves, and hopefully, you're now in a position to make a slightly more informed decision. ◻

LEARNING TO SOLDER

If you're new to electronics, soldering is a great skill to learn. The equipment is cheap and widely available, and you can learn the basics quickly. There are great introductory videos from Raspberry Pi (**hsmag.cc/zeEgKd**) and Adafruit (**hsmag.cc/g5S0vl**).

SUBSCRIBE TODAY FROM ONLY £5



SAVE UP TO 35%

Subscribe today and get:

- **FREE delivery**
 Get it fast and for FREE
- **Exclusive offers**
 Great gifts, offers, and discounts
- **Great savings**
 Save up to 35% compared to stores

Subscribe online: hsmag.cc/subscribe

SUBSCRIBE TODAY

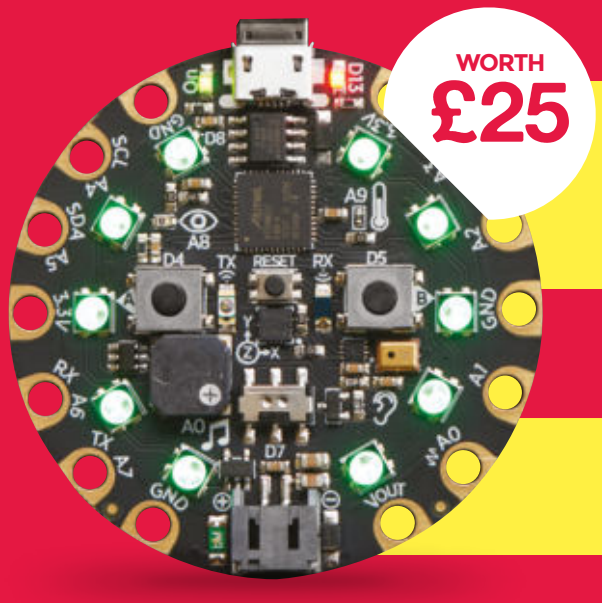
Subscribe for 12 months | **Rolling monthly subscription**

£55 (UK) **$60** (USA)

£80 (EU) **£95** (Rest of World)

Free Circuit Playground Express with 12-month upfront subscription only (no Circuit Playground Express with rolling monthly subscription)

- **Low monthly cost** (from £5)
- **Cancel at any time**
- **Free delivery to your door**
- **Available worldwide**



FREE!

Adafruit Circuit Playground Express

With your 12-month print subscription

This is a limited offer. Offer subject to change or withdrawal at any time.

SUBSCRIBE on app stores

From £2.29

Available on the App Store

GET IT ON Google Play

Buy now: hsmag.cc/subscribe



FEATURE

How I Made FACIAL RECOGNITION BAFFLER

Technology vs Technology – Face-changing projection mask

By Sean Hodgins

Let's face it, facial recognition as a technology is here, and at this point, its implementation is trivial. Anyone with a Raspberry Pi, a bit of experience in Python, a webcam, and an hour to kill can get a basic example up and running.

FACIAL RECOGNITION WITH PYTHON AND OPENCV

It's fun to play around with, and can be a really great lesson in object detection and, in some examples, machine learning. Like all technology, it is not inherently evil. Its virtues and pitfalls are all a matter of how it is and will be used. There are plenty of positive examples of facial recognition tech. Most involve combining security and convenience in some way: unlocking your phone or house, giving you access to your bank account, and generally securing things that you only want a select group of people to have access to. However, there are also a number of concerns that arise from widespread government and state use of facial recognition. These concerns range from the relative banality of targeted ads, all the way to dystopic *Minority Report*-type situations. Many of the worries about this kind of technology stem from concerns about protecting personal privacy.

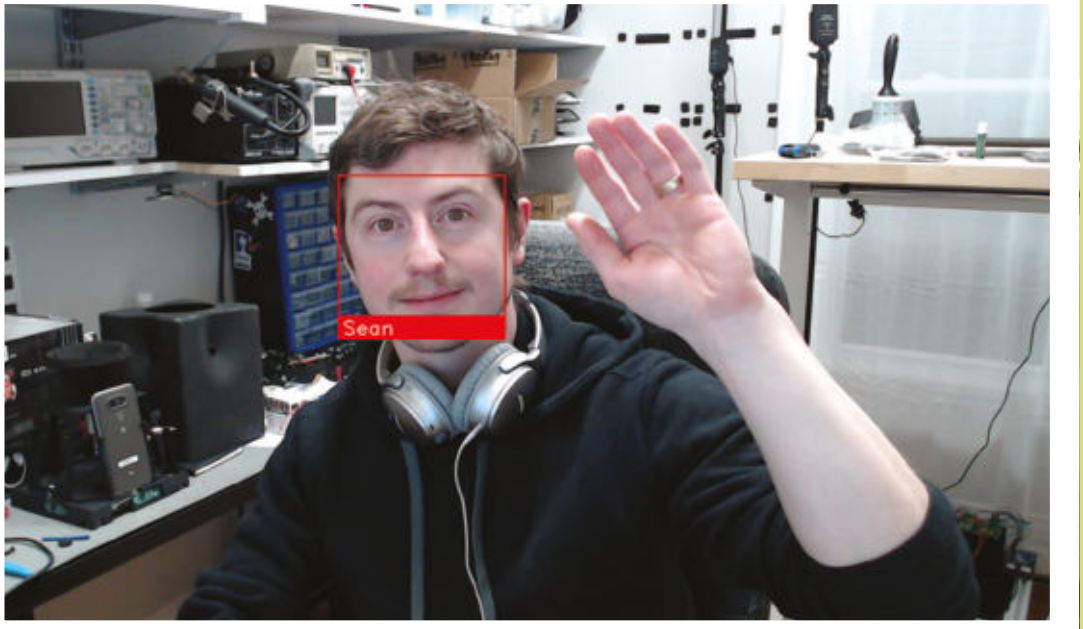


Above Be anyone you want to be

If you knew every street corner had a camera logging your every movement (like what your phone is probably doing right now), you would feel that the government is invading your privacy in some way. Well, I have bad news for you. It's already happening…

MY OWN FACE PROJECTED ON 'THE MASK'

Fortunately, I do have a solution for you; it's called 'The Mask'. The Mask lets you be whoever or whatever you want. Change your face at any time. It works by combining



Right
Computers can tell who we are

a small projector with a Raspberry Pi Zero, and using them to project any image you want onto the surface of the mask.

DIFFERENT IMAGES PROJECTED ONTO THE MASK

OK, let's get real for a second. I'm not proposing that this dynamic projection mask would work any better than its static counterpart when it comes to hiding your identity. But let's do a fun thought experiment. Imagine a system that is identifying, tracking, and logging every single public individual's face it comes across, and assigning them to an ID. Instead of just masking your face, you could trick the system by distorting or altering your 'face' to become a different person. Rather than simply being unrecognisable to the system by wearing a static mask, you could mess with the data collection protocol by wearing The Mask and create erroneous data, false inputs, and confusion. Wouldn't that be fun? Well, if you've got a 3D printer, fabrication experience, and some very minimal soldering skills, you can!

Let me start off by saying this isn't going to be a 'How-to' article. I tend to not do step-by-step DIY articles and instead give you general concepts; reasons why my brain thinks a certain way, or why I made specific decisions in the build. I will also share with you some mistakes I made along the way. I'm basically just steering you in the right direction. Before we get into it, I'm going to suggest you check out the YouTube video I made on The Mask, so you can see how cool it is and how it performs in real life. Check it out here: **hsmag.cc/KLcAhl**.

When I start a project, I often already have an image of how the finished product will look in my head. When I saw this, I imagined it to be futuristic and cyberpunk-ish. Something different. I thought low poly would be cool because it's like a bunch of smaller flat projection screens. Not to mention that it would probably be easier to create projection maps for the flat panels, instead of a smooth face. I've come across low poly masks before. I've even made a few out of paper. However, kongorilla's mask on Thingiverse (**hsmag.cc/t5dTAC**) has shown up in a few projects I've seen over the years, and I knew it would be perfect for this one as well. The other components (the projector, Raspberry Pi, and battery) were all chosen out of basic necessity and usability. Deciding between the TI DLP LightCrafter projector and a generic battery-powered projector wasn't an easy choice – they both have positives and negatives. An inexpensive off the shelf battery-powered projector unit could easily connect directly to an Android phone and display anything from it. Ultimately I chose Raspberry Pi and the LightCrafter because of the potential for creating and adding cool features in the future.

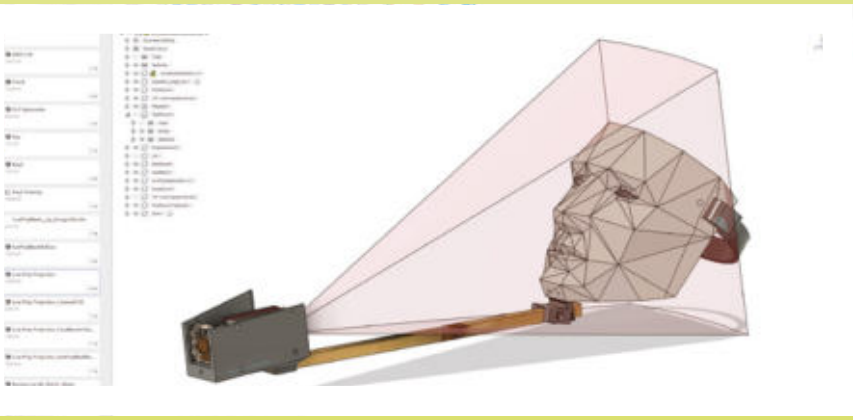
I started out by importing the mask, Raspberry Pi, and projector into Fusion 360. I like to browse GrabCAD before I create my own components because often someone has already done an equally good, if not better job, designing something, which can →



Right
The Mask is also useful for Kryten cosplay

FEATURE







Top
A projector shines the face onto the mask

Above and left
Calculating the geometry of the image

save a lot of time. The projector is mounted on its side (portrait mode) to utilise more of the resolution available. At this point, there were some options in the design. The projector could either be mounted to the top of the mask with an angle facing down, or to the bottom of the mask facing up. Facing down would give a better image and be easier to project onto, but would be more in the way from others viewing the mask and more difficult to walk around with. Facing up would be less obtrusive and straining to wear, but distort the image more. In the image above, you can see I tried to estimate where the projector will lay on the mask, so I knew how far it had to be from the mask. I should mention, I originally had plans to mount the projector on my chest and track the mask's location, but decided to scrap that idea to simplify the project, and I'm glad I did. Optics could also be used to shorten the throw of the projector, but in terms of timeline (I wanted to finish before Halloween, which at the time was two weeks away) and complexity, I stuck with the stock lens.

I decided to mount the electronics to a square wooden dowel and attached this to the chin of the mask. I went through a few





Above
Housing the electronics

Left
The projector is protected by its enclosure



Above
Set up and ready for testing

iterations of this 3D-printed part until I had something that was structurally rigid enough to support the electronics module without bending or moving around. Rigidity ended up being a very important part of this build, and can throw off the projected image with very little movement. Pro tip: make more modular parts, instead of one large printed unit. Large prints can often take such a long time. Save yourself some time by printing smaller parts, so individual pieces can be modified and screwed together faster. The housing for the electronics was designed to be as small as possible, and I included a small fan to circulate the air as well, which wasn't necessary in the end since the projector didn't generate that much heat. It might be useful in the summer. The mask itself was mounted to an inexpensive welding helmet strap. It works very well, and I could see myself using this method for mounting other masks in the future.

CHANGING FACES

The mask can be easily printed with no supports. On a scale from 1 to 10 on the difficulty of the electronics compared to the average project I've done, this is a solid 1 (least difficult). It's essentially just a pinout adapter board. The projector comes with a pinout that is supposed to connect directly to a BeagleBone Black. Since this project uses a Raspberry Pi Zero, the pins on the projector need to connect to the appropriate pins on a Raspberry Pi instead. It just needs two sets of headers, and for Raspberry Pi Zero, I soldered directly to the board. I did include some circuitry to monitor the 5V line, but again, didn't end up using it. I just bridged the 5V line directly to Raspberry Pi, and that was that. There is also a fan control circuit, a MOSFET, but if you're not going to use the fan, then that can also be left off. No surface mount components need to be soldered to the circuit board. The power cable is a 2.1mm right angle barrel jack to USB-A. I zip-tied the cable to the dowel and plugged it directly into a power bank. It hangs down the collar of the sweater and is hidden from sight. Having it down the sweater relieves some of the weight of the mask. Once you have everything 3D-printed, soldered, and assembled, it's time to take on the next and equally complicated part: making the projections for the mask.

This requires you to know how to edit and create videos, and use Photoshop to manipulate facial images of people. Since the mask is angled back to the projector, everything is skewed in the same dimension. After some trial and error in Photoshop, I figured out the perspective, and I was able to quickly grab images of people from Google and convert their faces to the mask. There is a bit of a process to get something projected onto the mask, and there are several ways to do it, both programmatically and manually. My method involved manually creating a video that was the same resolution of the projector in After Effects, but with blacked out areas where I did not want anything projected. I would create the content I wanted, export the video, and send it to the Raspberry Pi through a VNC program. Once the →



Above
It's a shame they never made a *Matrix* sequel



Left
Sean, meet Sean

video file was on the Raspberry Pi, I would be able to SSH into the Raspberry Pi from a phone and run a command using OMXPlayer that would play the video on the projector. I found an app to simplify this by having buttons with predefined commands. The app is called RaspController for Android. You can connect a Bluetooth speaker for sound. Raspberry Pi would struggle a bit with both the Bluetooth and WiFi running at the same time, and I didn't have the best results with doing it this way. I want to implement something on the hardware level in the future.

In order to complete the look, you need to wear a dark black hoodie, for two reasons: firstly, this makes you look like the stereotypical hacker, and secondly, it hides all of the black electronics in front of you when you stare at people. It's a really great effect and works surprisingly well.

If you want to project your friend's face on the mask, take a photo of them from the same angle that the projector is to the mask. This way, you avoid having to do any serious Photoshopping to the image. The effect works great with video as well. I was able to project my own face talking onto the mask. It's a pretty creepy effect, and something I did not show in the video, but might release some more effects in the future. Going forward, I can see myself adding a camera built into the electronics that could capture and steal someone else's face instantly. This was the main reason I chose to use Raspberry Pi in the first place. I would also like to shorten the throw of the project so that it's more compact and easier to wear.

I was able to project my own face talking onto the mask. It's a pretty creepy effect, and something I did not show in the video

In a world where technology advances much faster than our laws and policies can control or regulate, it's fun to create projects to get people thinking. In its current form, The Mask is not very practical to wear on a daily basis or out in the general public. This is one reason why I like to open-source my projects. Other makers might take it and find a way to expand or improve on the design in a way I didn't see. If that's you, do it!

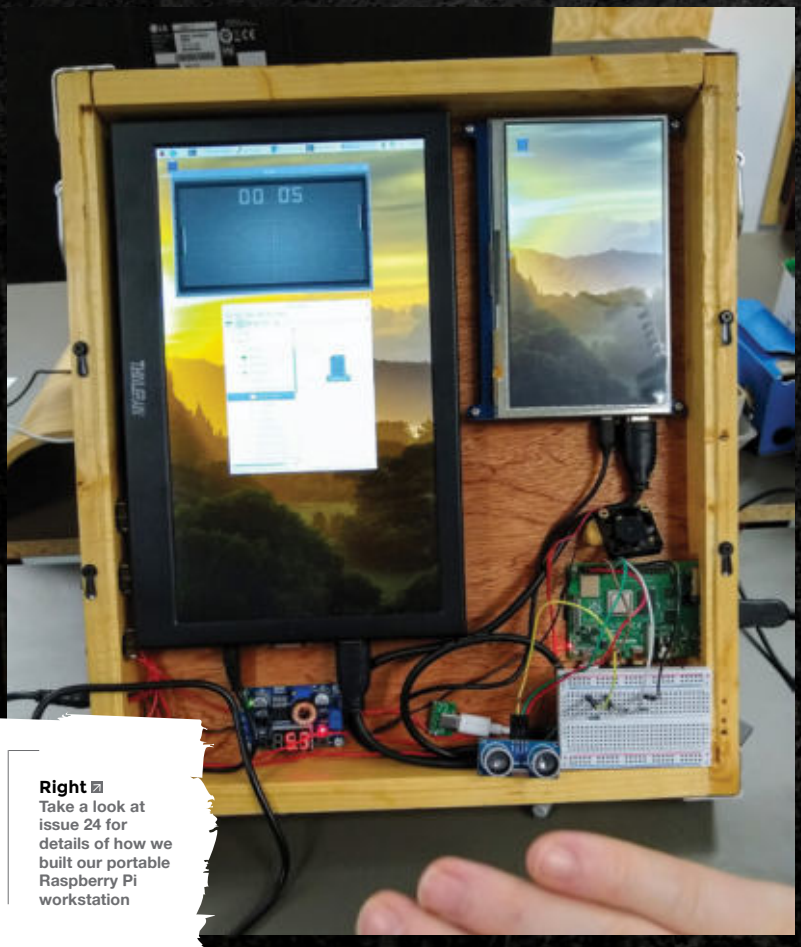
If you would like to recreate this project, it's completely open-source, and all of the resources are available online. The easiest way to find them is to navigate to the description of the YouTube video mentioned on page 53. I also posted a series of smaller examples of cool effects the mask can do on my Twitter account **@idlehandsdev**. I often share extra titbits of information and behind the scenes photos of my projects on Twitter, so if you want to catch those, give me a follow. If you like these kinds of open-source projects, subscribe to my YouTube where I am always working with Raspberry Pis, Arduinos, and doing custom builds.

FEATURE

IN THE WORKSHOP: Ultrasonic Pong

By **Ben Everard**

Play this classic game without touching the keys



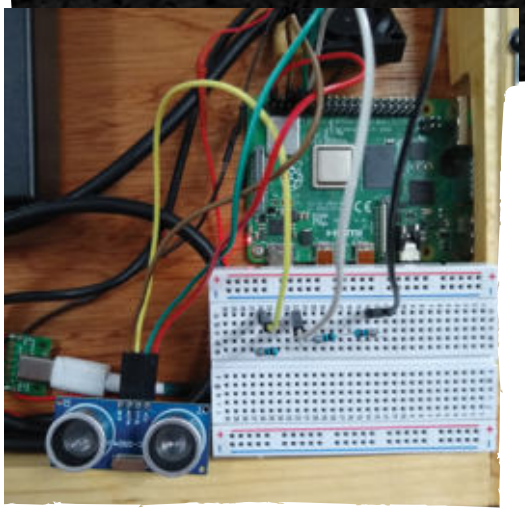
Right
Take a look at issue 24 for details of how we built our portable Raspberry Pi workstation

You might have noticed that Raspberry Pi Press – the publication team behind HackSpace magazine – has released a new book called *Code the Classics*. In this, a team of authors, coders, and artists have recreated some of the great video games of the past. This is an entertaining book for a whole bunch of reasons, but one aspect of it that is particularly exciting to us makers is that it means there are some games out there that are really fun to play, but also written to be easy to understand and have high-quality game art to go along with them. Why does this excite us as makers? Because it makes them ideal candidates for testing out novel DIY games controllers!

We're going to start right at the beginning of the book (and also at the beginning of computer game history) with the game Pong. There's a great chapter on this seminal game in the book (which is available to purchase in print or download for free from **hsmag.cc/OD5eZp**), but we'll dive straight into the source code of our Boing! tribute game.

This code should run on any computer with Python 3 (and a few dependencies) installed, but we'll use a Raspberry Pi, as this has GPIO pins that we can use to add on our extra controller.

Download the code from **hsmag.cc/m3Xd7N** – just hit the 'Clone or download' button, and then 'Download ZIP'. Unzip the downloaded file, and you should have a directory called **Code-The-Classics-master**, and inside this a



Left
Using three of the same resistor type gives us the two-thirds step-down we need in voltage

directory called **boing-master**. Open a terminal and navigate to this directory, then run:

```
python3 boing.py
```

If everything works well, you'll get a screen asking you to select one or two players – press **SPACE** to confirm your selection, and have a play.

HACKING THE CODE

So that's how Eben Upton designed the game to be played. Let's put our own spin on it. Games controllers are basically just sensors that take input from the real world in some way and translate that into in-game actions. Most commonly, these sensors are buttons that you press, but there's no need for that to be the case. You can use almost any sensor you can get input from – it sounds trite, but the main limitation really is your imagination!

We were playing with ultrasonic distance sensors last issue, and this sprung to mind for a Pong controller. After all, distance sensors measure in one dimension and Pong bats travel in one dimension.

Last issue we learned that the main challenge when using the cheap HC-SR04 sensors with 3.3V devices is that they use 5V, so we need to reduce their output to 3.3V. A simple voltage divider does the trick, and we used three 330Ω resistors to achieve this (see **Figure 1** for more details).

There's support for these sensors in the GPIO Zero Python library. As a simple test, you can obtain the distance with the following Python code:

```
import gpiozero
import time
sensor = gpiozero.
DistanceSensor(echo=15,trigger=14)

while True:
    print(sensor.distance)
    time.sleep(0.1)
```

That will give you a constant read-out of the distance between the ultrasonic sensor and whatever object is in front of it. If you wave your hand around in front of the sensor, you'll see the numbers changing from 0 to 1, which is the distance in metres.

So far, so straightforward. We only need to add a few bits to the code of our Boing! game to make it interact with the sensor. You can download an updated version of Boing! from **hsmag.cc/issue28**, but the changes are as follows.

Add this line to the `import` statements at the top:

```
import gpiozero
```

Add this line to instantiate the distance sensor object at the end of the file (just before `pgzrun.go()`):

```
p1_distance =
DistanceSensor(echo=15,trigger=14,queue_len=5)
```

We added the `queue_len` parameter to get the distances through a little quicker.

Finally, overwrite the `p1_controls` function with the following:

```
def p1_controls():
    move = 0
    distance = p1_distance.distance
    print(distance)
    if distance < 0.1:
        move = PLAYER_SPEED
    elif distance > 0.2:
        move = -PLAYER_SPEED
    return move
```

This uses the rather arbitrary settings of 10cm and 20cm to define whether the paddle moves up or down. You can adjust these as required.

That's all there is to our ultrasonic Pong. It's great fun to play, but there are, no doubt, loads of other versions of this classic game you can make by adding different sensors. Why not see what you can come up with? □

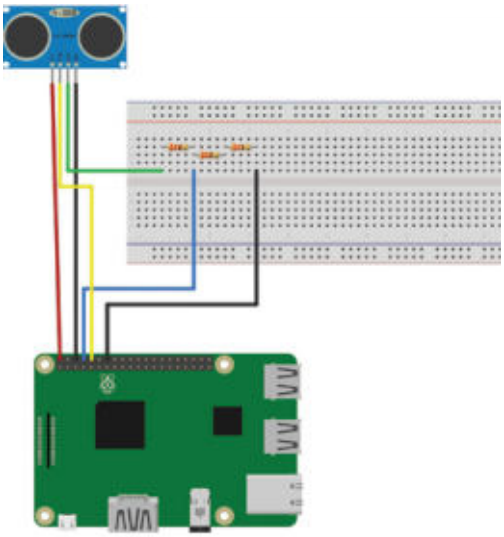


Figure 1
You only need three resistors and one sensor to build this games controller


CODE
THE
CLASSICS
VOLUME 1
Brimble
Crookes
Gillett
Malone





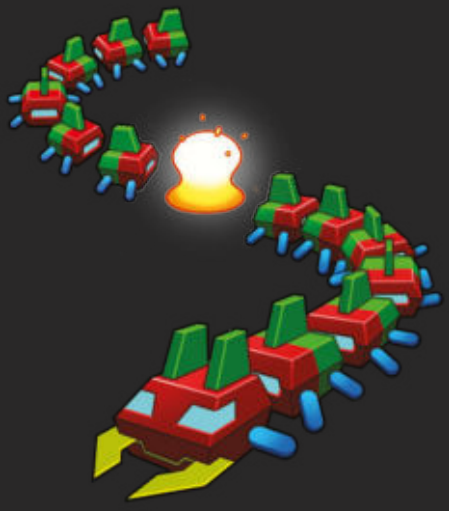
CODE THE CLASSICS
VOLUME 1

This stunning 224-page hardback book not only tells the stories of some of the seminal video games of the 1970s and 1980s, but shows you how to create your own games inspired by them using Python and Pygame Zero, following examples programmed by Raspberry Pi founder Eben Upton.





- ***Get game design tips and tricks from the masters***
- *Explore the code listing and find out how they work*
- ***Download and play game examples by Eben Upton***
- *Learn how to code your own games with Pygame Zero*



Available now **hsmag.cc/store**

HackSpace magazine meets...

Dr James Hilder

When the robots rise up, it's thanks to this man that they won't immediately kill us all. Probably

Dr James Hilder is a senior researcher at the University of York, looking into safety systems on self-driving cars. As well as his own research, he's also heavily involved in making robotics kit for the students to work on, taking masses of sensors capable of gathering huge amounts of data, and turning them into kits that can be used to explore the maddeningly complex world of robotics. We ventured over to the York Robotics Laboratory to find out what, in the words of Tom Waits, he's building there. →



Above
Swarm robotics, safety, autonomous rescue bots – all just a day's work for a robot wrangler

HackSpace We're surrounded by kit in here. You must really love making things?

Dr James Hilder I've got boxes and boxes of gear. I must have everything that Pimoroni and Adafruit make. I love the whole getting computers to interact with hardware. That's my passion. Bridging between the two. And I think it always has been. When I did my undergraduate degree in electronics, I always liked the fact that I was doing a bit of computers in there, but making them interact with the real world is where the fun starts. Since the Mbed Device Platform – that predates Raspberry Pi, and came out at about the same time as the first Arduinos. Like the Arduino, it doesn't have a full operating system on there.

When it came out, it was £40, but the downside is that it's still £40, when Arduino and even Raspberry Pi Zero are so much cheaper. The compiler is online, like Arduino, and it has all sorts of inputs that make it useful.

All of a sudden, you could interface with things. Before 2008, if you wanted a PC to turn an LED on, it was really hard to do. To have a user-defined LED or to read a voltage, simple things like that were really hard to do. Boards like this came out and changed everything. Raspberry Pi was the next step: it included everything on one board, so you didn't need a computer; the board was the computer.

HS That seems like a flagrant breach of Moore's law for that to be the same price as it was in 2008.

JH There's a lot of that about. And that's why we make a lot of our own robots.

The most widely used swarm robotics platform is the e-puck, which is about twelve years old.

You can still buy them, and they sell for about £1000. And that's why we end up making our own robots. This is the weird world of research robotics: add a zero to what something actually costs, and that's what you end up paying. They were developed at EPFL; they're a Swiss organisation. There are probably half a dozen big robotics institutes in Europe, and EPFL is definitely one of them.

Anyhow, a couple of years ago, we made an extension HAT that took the e-puck and stuck a Raspberry Pi Zero on it, which fits pretty much perfectly. It's almost as if they designed the e-puck with the thought that ten years later, Raspberry Pi were going to release this £5 computer.

We also make this adaptor shield. By changing the orientation of the board, we can fit a larger Raspberry Pi in there. You find that a lot with the e-puck: loads of people do extensions for it, so you end up with these really top-heavy robots.

"The most widely used swarm robotics platform is the e-puck, which is about twelve years old"

HS That's crazy, that something so simple can be so expensive.

JH Price does scale with simplicity. For example, there's a tiny, absolutely minimal robot called the Kilobot. They're these little tiny robots – they have a coin cell under the PCB, three metal legs, of which two vibrate. By controlling the rate of that vibration, you can make them crudely move in a pattern. They have a single infrared photodiode on the top and the bottom. You can do one really cool experiment with them, and that's it. You write code for them, and it makes them want to cluster, to aggregate and form shapes. And you can change the code so they don't want to cluster and they break apart, and it looks amazing. But that's the only thing you can do with it. When you've done that, you've got 1000 robots that do only one thing.

Harvard published a paper on this, and they made a really big thing about it only costing $14. And then other universities wanted their own swarms of robots, that do only one thing, because it sounds good to be able to say, "we've got a swarm of 1000 robots". Instead of 20 e-pucks, you could have 500 of these things. But the trouble is that you can't do anything with them. You can't publish any papers, because you can't do any experiments with them, beyond what has already been done.

It's the same, to an extent, with the e-puck: there's the firefly algorithm where you synchronise the robots using pulses of lights: one will pulse at random, then the others will detect that and start to pulse, and eventually they'll all have a coherent pulse.

It's incredibly low level. It's a way of synchronising things among a number of different units, and that's it. People have ever so slightly adapted that, so you can do primitive counting by the number of robots that are in a cluster, or really, really simple things like the algorithm that's inspired by what bees do. There's a lot less going on in the field of small robots than there was a few years ago.

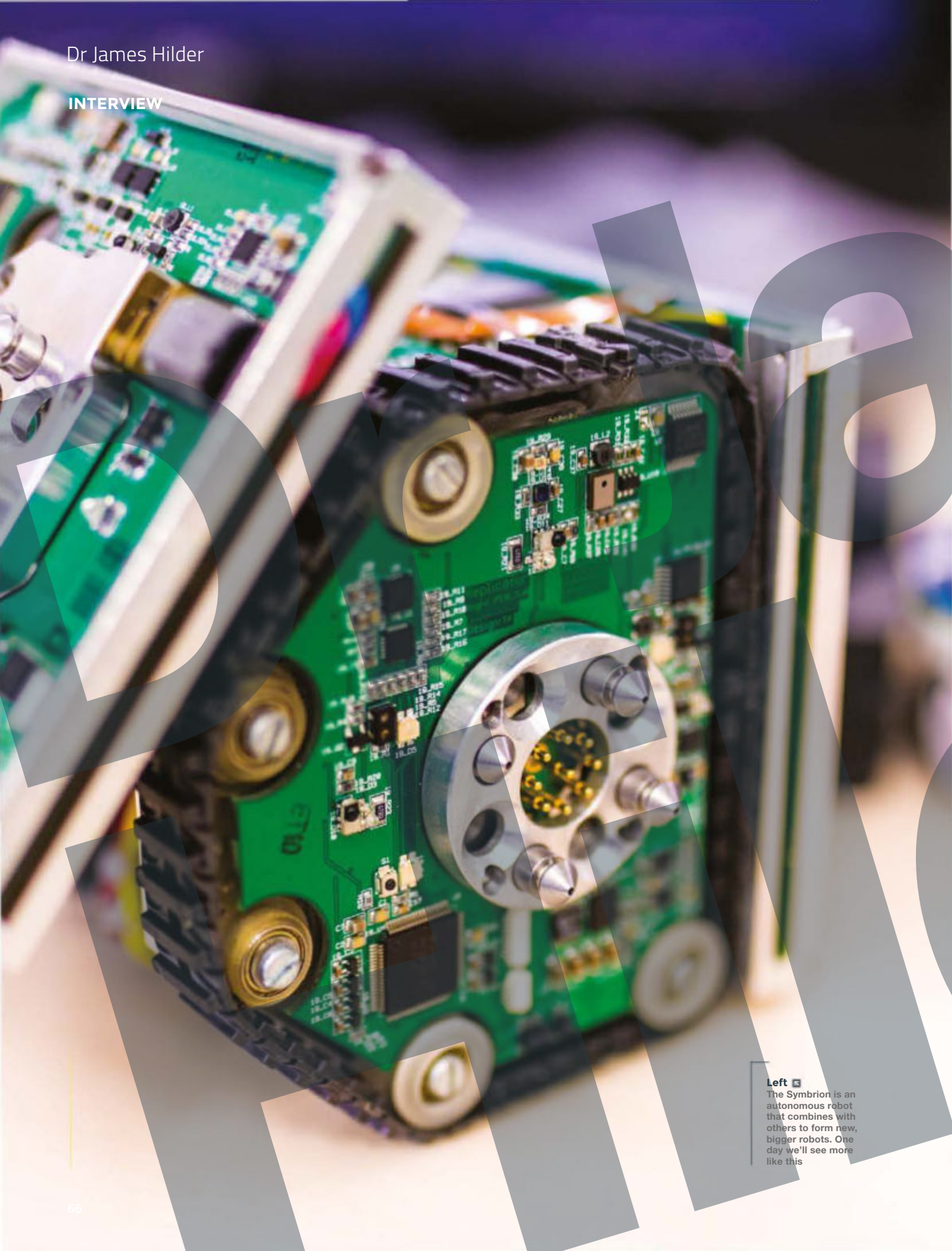
Everything in academia in the last ten years has become based on how we're going to make money out of it. Is this going to produce some patents that will make us rich?

HS So what are your main areas of research here?

JH We're building a massive new robotics lab, and one of the things I'll be looking at is how to make driverless cars safer. There are two fields in driverless cars: one is ethics: when something goes wrong, who do we crash into? And the other is safety: who do we sue when the inevitable does happen and someone gets killed. Is it the programmer? →



Left
At £200 each to make, these are massively cheaper than the standard university research robot, and it's thanks to open hardware that this is possible



Left
The Symbrion is an autonomous robot that combines with others to form new, bigger robots. One day we'll see more like this

The CEO? The shareholders? The person who wrote the operating system?

The building will still be used for teaching. We do a one-year MSc course in intelligent robotics and, over the last four months of that course, the students do a group project based on a search and rescue vehicle.

At the moment we've got a group of students in here from computer science who are working in their spare time, working on a robot for Pi Wars. They're working on a robot that does a number of different challenges.

We also have a Raspberry Pi Challenge here at the CS department. Every year it used to be the case that, in summer before each cohort arrived, we would send them each a Raspberry Pi. Then, in the first week of term, they have this competition, to play some AI game against the other students.

We're using Raspberry Pis very much more now that we're moving away from the swarm stuff, with something called ROS, the Robot Operating System. ROS is very big now in robotics. [The] first thing to know is that it's not an operating system: it's middleware. You can install it onto a Raspberry Pi with apt-get, and it's a set of libraries that package up communication protocols.

Say, for example, we have a sensor and we have ROS installed, we could write a ROS service that takes the data and pushes it, and that can be read by any machine that's got ROS running, that's connected either to the network or on the robot. And it extracts away all that low-level stuff to do with interprocess communication. It's very handy if you have multiple microcontrollers or Raspberry Pis. You might have something with a higher-end vision processing thing.

When we're dealing with self-driving cars, most of the machine processes that we're dealing with – the machine vision probably isn't done on Raspberry Pi; it's done on something a bit beefier, such as an FPGA, or it may be farmed off to the cloud to do the vision processing.

But then, ROS provides this framework so that Raspberry Pi can read the important data from that: 'Is there an obstacle in my way?', 'Can I see a person?' Whatever it is you're looking for.

HS Can you tell us about the York Robotics Kit that you've made to work with Raspberry Pi?

JH We started that in September, so not long ago. Every year the students come in and, as part of their group project, they have to make a robot that will rescue a teddy bear that's had a road traffic accident. Find the car and the teddy, take an aerial picture using a drone, send a land-based robot to pick up the teddy, and take it back to the base. If they want full marks, it has to be autonomous: press one button, and it just happens.

> "As part of their group project, they have to make a robot that will rescue a teddy bear that's had a road traffic accident"

Every year the groups make their own land-based robot. The course has changed to incorporate a lot of electronics, robot design, making an H-bridge, which is good. But they only have a limited amount of time. They have to do all these tricky things like localisation (knowing where a robot is indoors), which is really tricky.

But then the trouble is the learning process of learning how to make anything before you make it do things. It's fine if you're just adding a motor driver, but once you add all the other bits on, it gets complicated. So, I thought I'd speed up the process and save having to make PCBs for each project – let's make a HAT for a robot that augments a lot of what you can get from each of these Adafruit modules. They do a lot of things, but I haven't seen one HAT that puts everything all together.

Obviously, your robot needs a power supply of some sort, because you've got to go from a battery, usually a LiPo battery, down to something that can power a Raspberry Pi, and also power the motors. Then, you've got your motor drivers; invariably we're dealing with brushed DC motors, so you need to make sure you've got the power right there. Then there are the sensor inputs. Obviously, there are loads of different sorts of sensors. Things like the ultrasonic ones that cost a couple of quid. These are almost ubiquitous – they're good at what they do, they're analogue, they're reliable. But obviously, your Raspberry Pi doesn't have ADC (analogue-to-digital converter) ins, so we need an ADC on there.

We started getting really interested in the idea of sensor fusion. Sensor fusion is where you take data from multiple sensors and run some sort of algorithm on it, things like neural networks, to make the data more robust or more useful or more reliable. We're interested in that because the trouble with different sets of sensors is that they will react differently in different conditions. One distance sensor we use doesn't return anything when there's an object right up next to it, and doesn't return anything when things are too far away; it also gets confused by windows.

There's a different kind of proximity sensor that's used in mobile phones to detect when they're close to your face – if you can combine the two, the inputs from both get a lot more useful. Then we have a longer range one that isn't so great under specific light conditions. Add that one as well, and you're expanding the usefulness of each one.

What you can do is have an algorithm that takes input from all of them. If my algorithm takes input from an ambient light sensor and tells me that the robot is in broad daylight, then I can take the →



Above
It's nice to know that university robotics labs use the same Adafruit and Pimoroni gear as everyone else

data from the laser distance sensor with a pinch of salt.

But the trouble is that all of these sensors use I²C. They're usually very small, and you can't choose what their I²C addresses are. So you need an I²C multiplexer. And we use microphones on the robots, so you need the right bus for that, which is I²S – despite the name, that's nothing to do with I²C. And we use PWM to give audio feedback, which also needs its own connector.

So, all these things are in the Robotics Kit. I've made ten of these in the last couple of weeks. It's right on the cusp of being a job big enough to outsource. Ten is tricky. We've got eight students this year, and all of the boards we're using are in electronics; the students are building their own robots around them.

We've split the design up, so it's on two boards. There's a power supply, there's an ATmega chip on there, [so effectively an Arduino], a soft power button, and it's got a temperature-based fan, because it's designed for Raspberry Pi 4, and Raspberry Pi 4 does get hot.

It's actually quite a lot of work to get a company to make these – there are so many components on it that you have to hold their hand a bit, give detailed instructions on certain bits, and maybe redesign the PCB to make it more obvious which way around components go.

Knowing the hardware means we can provide students with the software so that they can get started straight away. One of the most time-consuming things in the past was always getting the software set up, particularly with ROS and OpenCV. OpenCV, if you build it from source, takes ten hours on a Raspberry Pi Zero. There's a lot of complexity in the controller, and we bolt-on a simple, cheap computer. It works.

This is exactly what Raspberry Pi is for. Simplicity and cheapness, replaceability – all of these things are important. Because you can guarantee that robots will fail.

HS Failure's good, right? As long as you learn from it.

JH We've got a good example of that with this modular docking robot that came out of a massive EU project that ran from 2008 to 2013, called Symbrion. It was meant to be two different projects that the EU insisted were merged into one. The Symbrion part was supposed to be individual robots that could work on their own, or work as a swarm. But it also has connectors too, that they can dock together; what that means is that you can make advanced robotic organisms. Link a few of them in a straight line, and you've got a robot that can move like a snake. Dock 16 of them together in a capital H-shape, and you've got a robot that can walk, with a body and four legs. The interesting bit about this is that there's no one computer in charge – they are 16 individual entities that talk to each other.

> "The downside was that these robots discharge battery significantly faster than they can charge up"

We were working on a field of research called artificial immune systems, and we were really interested in what happens when one of these individual robots fails – identifying faults and coming up with ways to react when something fails. What happens when the robot that's in the middle of this organism fails, or the battery goes – how do the others identify that it's gone, get rid of it, and either repair it or recruit another one? The algorithms in this were brand-new, and the researchers were trying to do it with brand-new hardware, and it's the most over-engineered hardware you've ever seen. It's incredible. Each face of the cube has a PCB, with not less than six microcontrollers on it. It is the best example of too many cooks. But that's typical of EU-funded projects: they're really ambitious, they're really big, you wouldn't get funded unless you had a big idea.

The EU also like grand challenges, something big that the robot can demonstrate at the end of the project. The grand challenge in this is that they wanted to have 100 robots alive for 100 days. So there's going to be an arena like this, where they put 100 robots in and they would get them to do some task, and still be running after 100 days. There would be a limited number of charging points, so the robots would have to monitor battery levels, and go and dock to recharge when batteries get low, and be able to share charge between them.

It's hugely clever. The downside was that these robots discharge battery significantly faster than they can charge up. Let's just say that the challenge was not successful. We've got a PhD student working on a very similar thing here; it's going to be brilliant. Watch this space. ◻

Below
James describes this as the smallest robot it's practical to make with the York Robotics Lab HAT



FEATURE

IMPROVISER'S TOOLBOX

TOOTHPICK

Use the pointy wooden stick to do a lot more than just pick at your teeth



Mayank Sharma

@geekybodhi

Mayank is a Padawan maker with an irrational fear of drills. He likes to replicate electronic builds, and gets a kick out of hacking everyday objects creatively.

We've been rummaging through the improviser's toolbox for over two years now, but nothing has blown us away like the toothpick. It's easy to be deceived by its simple looks; after all, it consists of a single part made of a single material, and was intended for a singular purpose that gives it its name. Yet, it's been used everywhere from the dining table to the battle-field, and everything in between.

The humble toothpick has the most colourful history and, surprisingly, a very secretive present. In fact, Henry Petroski, a professor of civil engineering and history at Duke University, has written an entire book on it that traces the history of the device that predates our arrival! Petroski writes that anthropologists have found evidence of their use in fossilised skulls of Neanderthals.

> "Petroski writes that anthropologists have found evidence of their use in fossilised skulls of Neanderthals"

Toothpicks are so ingrained in our culture that they were once used to ascertain a person's status. The royals and nobles picked their teeth with specially crafted toothpicks made from precious metals like gold and silver. The ancient Roman emperor Nero is said to have made his way to banquet halls chewing on a silver toothpick.

American inventors Silas Noble and J.P. Cooley patented the first machine for the manufacturing of toothpicks in 1872. Around the same time another American, Charles Forster, acquired the rights to a patent another machine, which he adapted to make toothpicks. Forster was a marketing genius who created a market for his toothpicks by employing Harvard students who would act in a displeased manner when the restaurant they just ate in couldn't bring them wooden toothpicks after their meals. Forster would then come in some days later to make the sale. Once they were readily available, diners started picking them up on their way out, and soon chewing toothpicks in public became fashionable. At its peak in the late 19th century, Forster Manufacturing was the world's largest producer of toothpicks.

Throughout history, toothpicks have taken on secondary usage. Even in the 16th century, everyone from nuns in Portugal to French restaurateurs was poking toothpicks in food; be it to hold sandwiches, or to pick out sticky morsels. You can find some usage drift for almost all the things in an improviser's toolbox, but the toothpick is in a league of its own. Some of its secondary uses have, in fact, given birth to completely new inventions, like cotton buds that were inspired by cotton balls wrapped around toothpicks. No surprise then that the toothpick was easy pickings for the maker community.

MINI TOOTHPICK DOLLS



Below
The build involves applying glue on some very small areas, and D'Andra suggests using a toothpick for this purpose as well

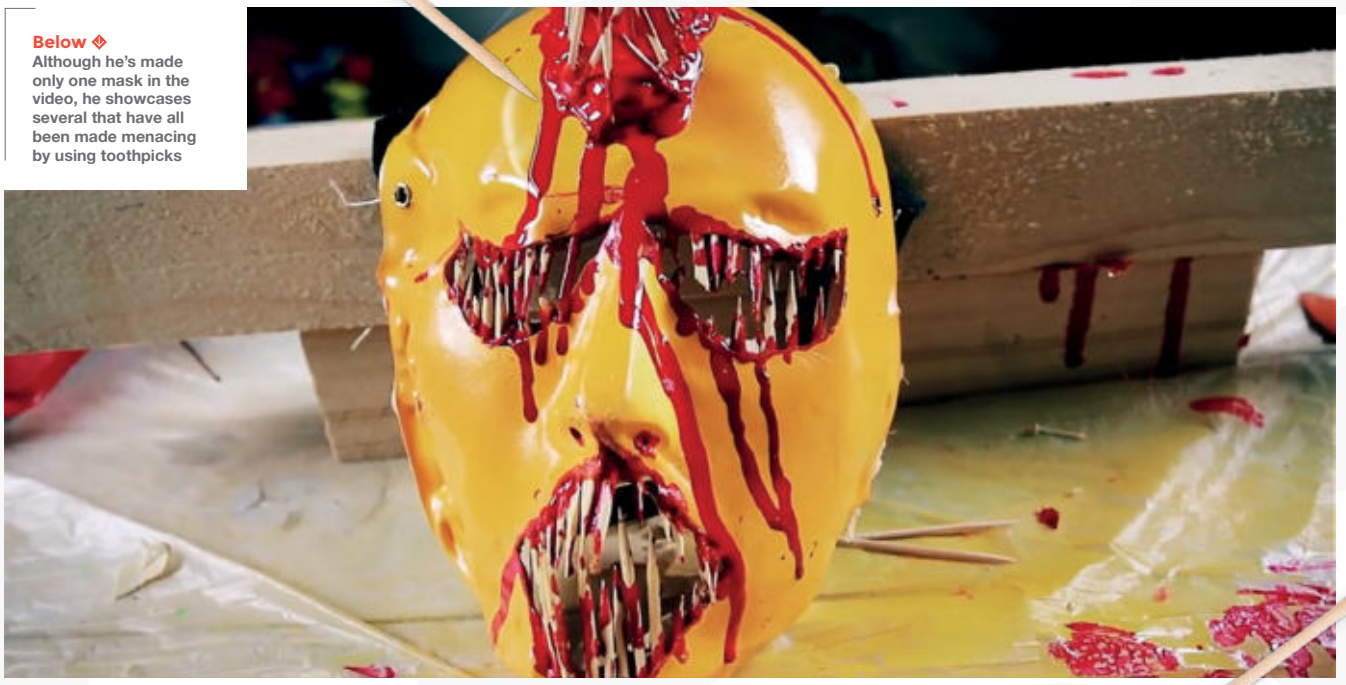
A self-confessed crochet fanatic, D'Andra Buchanan is a civil engineer by training, and an all-round DIYer by passion. As a mother of three, D'Andra is always on the lookout for things she can create with her kids. When she came across the toothpick dolls design in the Crafts n' Things magazine, she decided to build upon the idea with her six-year-old who she says, "wouldn't admit that he actually played with them for a little bit!" The build is fairly simple, and D'Andra estimates shouldn't take longer than an hour and a half. You'll first have to clip the pointy ends of the toothpick and cut some into smaller ones to create the limbs of the dolls. You'll then have to wrap them with coloured craft threads, and glue them together. She's used some sewing thread to create the dolls' hair, but you can use the craft thread as well. D'Andra has detailed the entire process for creating the complete attire for the dolls on her Instructables page. Before she glued the beads as the heads, she strung a needle with the thread that tied the hair together, and slipped it through the beads to further tighten the hair. →

"You'll first have to clip the pointy ends of the toothpick and cut some into smaller ones to create the limbs of the dolls"

Project Maker
D'Andra Buchanan

Project Link
hsmag.cc/EN2Apx

FEATURE



Below
Although he's made only one mask in the video, he showcases several that have all been made menacing by using toothpicks

HALLOWEEN MASK

Project Maker
MICHAL L

Project Link
hsmag.cc/zWCXXP

While D'Andra pruned the pointy ends of the toothpicks for her dolls, Michal used them to accentuate his Halloween masks. Michal is an intrepid DIYer who makes videos of creative and fun builds over on his Bonkers DiY YouTube channel. One of his projects involves mutating a harmless white mask into a scary Halloween wearable using a bunch of toothpicks.

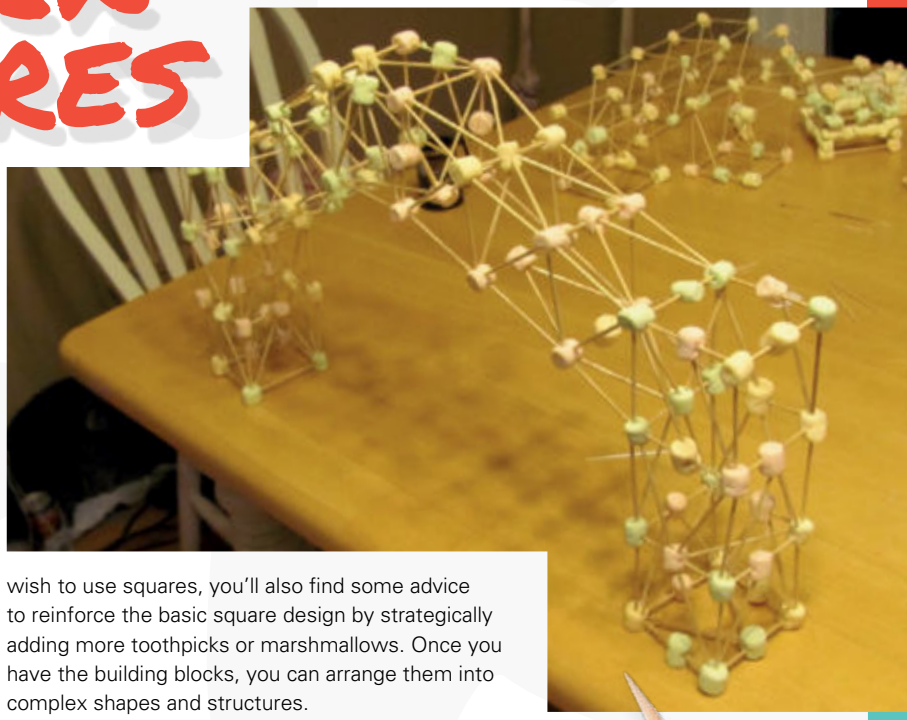
"WHEN IT'S DRY, HE PASTES THE TOOTHPICKS OVER THE EYES AND THE MOUTH ON THE INSIDE OF THE MASK USING HOT GLUE"

Michal first accentuates the eyes and the mouth of the regular mask before spray-painting it. When it's dry, he pastes the toothpicks over the eyes and the mouth on the inside of the mask using hot glue. He then dribbles red-coloured nail polish over the glued toothpicks to give the mask a sinister look. Michal suggests you can also use a small brush to apply regular paint instead of nail polish. He's used a decorative mask, which is why a major part of the build involves fixing a strap to turn it into a wearable one. However, you might find it a little more cumbersome to glue the toothpicks on the inside of such a mask. Remember to insulate the inside (Michal's used an old tee) to ensure you don't scratch your face while you're wearing the mask. If you recreate this, be very careful to make sure you can't get stabbed by the toothpicks.

MARSHMALLOW-TOOTHPICK STRUCTURES

Project Maker
Brian Clarke

Project Link
hsmag.cc/vRFDEB



You don't always need a Lego set to build things. Instead of using a traditional building set, Instructables user Brian uses toothpicks and marshmallows as his cheap and consumable rod and connector set. There isn't much to this build, but you'll find several useful tips and information to help with your structures. For one, he suggests allowing the marshmallows to get a little stale, which will give them greater strength and gripping power. Before you construct complex structures, you'll first need some basic building blocks. The Instructable dispenses some useful lessons in geometry to help you create strong structures. For instance, you are encouraged to arrange toothpicks and marshmallows into triangles, which are one of the strongest building blocks and inherently stronger than squares. In case you still wish to use squares, you'll also find some advice to reinforce the basic square design by strategically adding more toothpicks or marshmallows. Once you have the building blocks, you can arrange them into complex shapes and structures.

COCKTAIL UMBRELLAS



Project Maker
Emma Scott-Child

Project Link
hsmag.cc/DrkVp2

There's nothing quite like cocktail umbrellas to bring a tropical theme to any drink. Creative director and designer Emma Scott-Child will help you upgrade your drink using one really quick make.

You'll first have to trace a circle on a piece of colourful paper. For us, a circle with a diameter of about 4" turned out nicely. Now, use pinking shears to cut out the circle to get a scalloped edge. Then fold the circle into ten even segments, as she's shown in the video. After unfolding the circle, cut along the folds of one of the segments.

Then, overlap the cut edges and secure them with some glue. Now, cut ten toothpicks such that they are only a little longer than the radius of the umbrella, and glue them to the creases. Stick a skewer in the middle, and you're done. Pop in your drink and enjoy! ◻

FEATURE

3D printer strength

How do layer height, infill, and borders affect the strength of your prints?

By **Ben Everard**

If you need a 3D part to be strong, what print settings should you choose? Are there any hidden tricks or techniques that you can employ to ensure that your plastic stays in one place? In this article, we're looking purely at print settings, not part design, to see what changes you can make to ensure that you have parts strong enough for your needs.

What is strength? It's a surprisingly complicated question. Is it how much force it takes to crush an object? Pull it apart? Snap it? All of the above? Mechanical engineers define different types of strengths, including compression strength (how much squashing force can it take before it breaks) and tensile strength (how hard you can pull it apart before it snaps). However, in the real world, your parts are often subject to more than one force.

Our test was somewhat crude, but allowed us to deliver a consistent, repeatable force against an object. We printed identical bars, each with the HackSpace magazine logo on, and each with a hole in each end. We clamped these to a wooden frame so that the bar on the H sat on the surface, then tied a rope onto the hole and applied a force. We gradually increased this force and measured the deflection of the top of the bar from vertical. The force is measured in kilograms since we used a sprint scale to measure the force, but really, this should be in newtons.

We loaded the bar to 1 kg, then took this as the base reading. All subsequent readings are given compared to this base. Since we're measuring at the millimetre level, taking a base reading like this allows us to iron out any slight inconsistencies with the prints. While this method isn't flawless, the results did prove repeatable and consistent across multiple tests.

GETTING FULL UP

The first test we did was on different infill percentages. Each bar was printed using grid infill at differing percentages. You can see the results in **Graph 1**.

80% infill uses eight times as much material as 10%, so you should expect it to be stronger, and it is, but not proportionality. Despite this 8× increase in the amount of infill, there was only a 2.5× increase in strength. This isn't to be sniffed at if you need a strong part, but neither is it a magic elixir for producing strong parts.

For many parts, the actual breaking force isn't important, but the force it takes to deflect it. If you look at, for example, 5 kg of force, the differences between the 30% and 80% infill are all within the bounds of experimental error. In other words, if you need to prevent flexing at small to medium force levels, adding more infill may not help you.

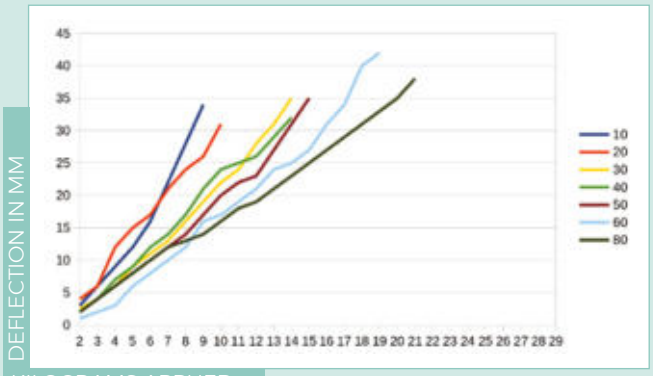
PERIMETERS

The second thing that we looked at was the effect of adding extra perimeters. This simply means that there are extra layers of solid perimeters around the side of the print. In this test, we used 40% infill, with increasing numbers of perimeters.

Graph 2 shows the impact this had. Here, as you can see, adding extra perimeters didn't make the structure significantly stiffer (even at higher loads), but it did mean that it would flex

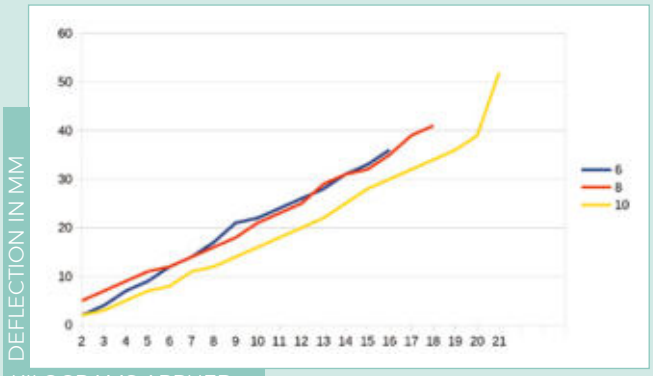
GRAPH 1

The deflection with force of bars with different amounts of infill



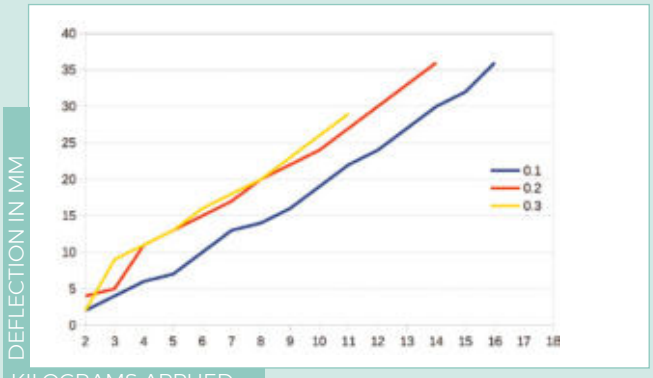
GRAPH 2

The deflection with force of bars with different numbers of perimeters



GRAPH 3

The deflection with force of bars with different layer heights



TEST KIT

All parts on test were printed on the same Prusa MK3S printer and came from the same roll of ColorFabb PLA Economy. In an effort to be as fair as possible, print speed, temperature, etc. were left at the default on the PrusaSlicer.

further before breaking. This differs from infill, where we found that the amount of flex at the point of breaking was fairly consistent, but the amount of force it took to get there was different.

Since we're also looking at two different ways of increasing the strength of the material, this begs the obvious question as to which is the most cost-effective. On average, each additional 10% of infill added 1.5 kg to the maximum force that could be applied before breaking. It also added 0.96 metres of filament (1.75 m diameter) to the print. This means that we required 0.64 m of filament to add an extra 1 kg to the force required to break the model.

Each two additional perimeters added 1.67 kg to the force, and only took 0.42 m of filament. This means that it only took an extra 0.25 m of filament to increase the force required by 1 kg (though, unlike the infill percentage, the number of perimeters has to be a whole number). Looking at this in isolation, it seems obvious that increasing the number of perimeters is the best option for increasing strength but, as always, it's worth testing on your particular part.

LAYER HEIGHT

Conventional wisdom in the 3D printer world says that larger layer heights are stronger. However, that's not what we found. We got consistent results showing that reducing the layer height increased print strength, by quite a large margin – see **Graph 3**. The print at 0.1 mm layer height broke at 25% more force than the print with 0.3 mm layer height. We're at a bit of a loss as to why this was, but it could perhaps be due to layer adhesion.

Selecting different layer heights does have a slight impact on the amount of filament used. At 0.1 mm, we used 7.98 m of filament, at 0.2 mm we used 7.93 m, and at 0.3 mm we used 8.60 m. This complicates things even further, because we would expect more material to mean a stronger model, but evidently, this isn't the case.

The only real conclusion we can draw here is not to trust accepted wisdom and test your parts if strength is important.

We've looked at just a few of the many, many options for tweaking 3D prints for strength. There are many more, and in future issues, we'll be investigating infill patterns, printing temperatures, print orientations, print speed, and more. □

A GRAIN OF SALT

3D-printing a strong part is a complex process, not least because strength means different things to different people. Your part may be under compression stress, tensile stress (stretching), shear forces, and others – and maybe more than one at the same time. The only accurate way of knowing what the best settings are for printing your piece is to print it with different options and test them out. Hopefully, we've given you some guidance of the right places to look, but our results shouldn't be taken as a general answer to all your 3D printing strength questions.

3 ISSUES from £5

on a quarterly subscription



Visit: hsmag.cc/345

HackSpace
TECHNOLOGY IN YOUR HANDS

FORGE

HACK | MAKE | BUILD | CREATE

Improve your skills, learn something new, or just have fun tinkering – we hope you enjoy these hand-picked projects

PG 84
WELDERS
Pick the right stick welder for your workshop

PG 86
RASPBERRY PI AUDIO
From beeps and buzzes, to high fidelity sound



PG 90
SHOP ORGANISATION
Make sure you know where to find your tools

PG 92
BELT DRIVE
Get the power from motors to where you need it

PG 78
SCHOOL OF MAKING
Start your journey to craftsmanship with these essential skills

78 Dithering
80 Boring

PG 96
PRINTED VASE
Missed Valentine's day? Redeem yourself with customised flowers

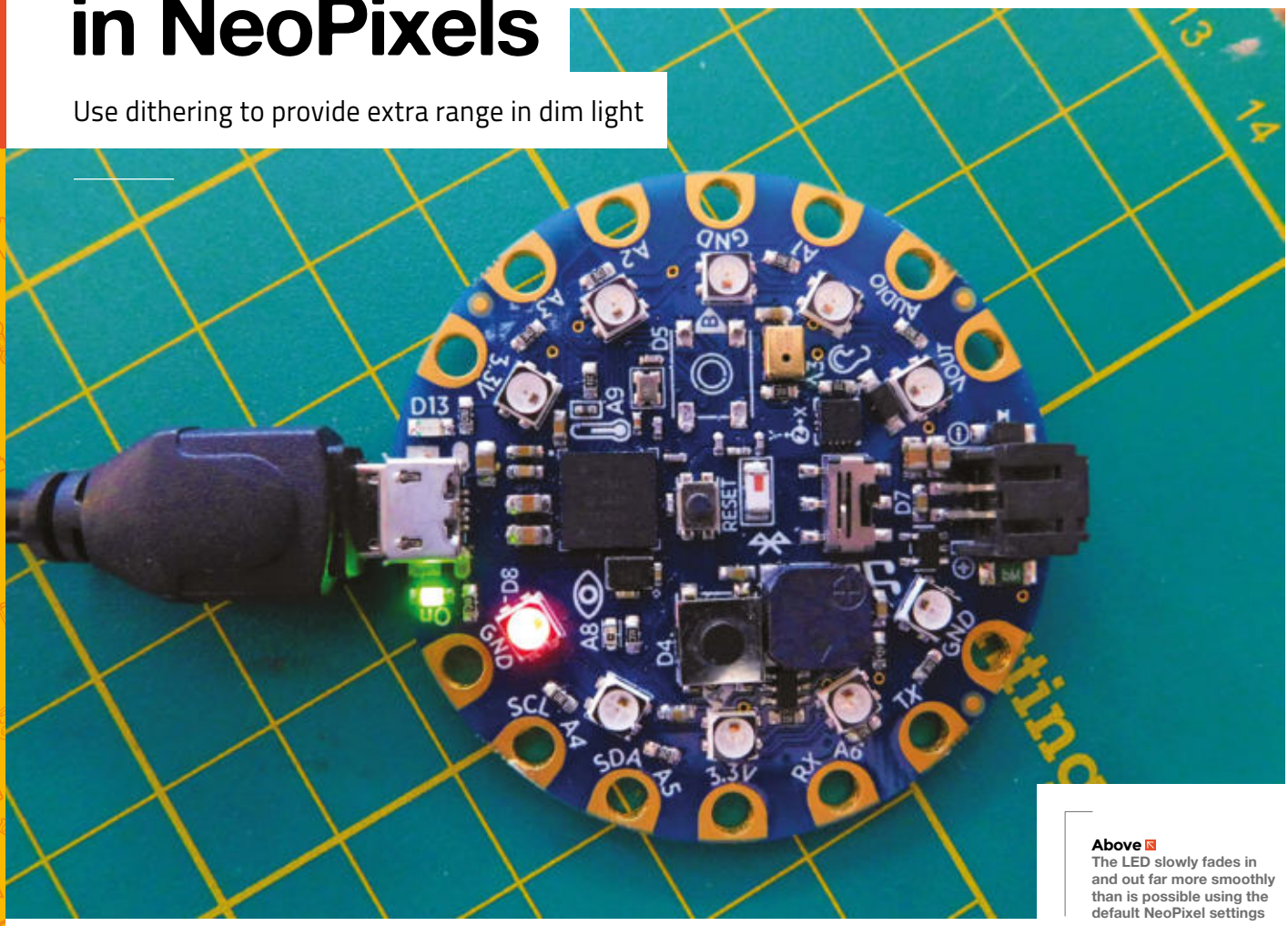


PG 104
BEAGLEBONE GAMEPAD
Low-latency IO using programmable real-time units



Getting extra range in NeoPixels

Use dithering to provide extra range in dim light



Above
The LED slowly fades in and out far more smoothly than is possible using the default NeoPixel settings



Ben Everard

@ben_everard

Ben loves cutting stuff, any stuff. There's no longer a shelf to store these tools on (it's now two shelves), and the door's in danger.

NeoPixels are great when you want to add colour – and interactive colour – to a project. They only need one pin on your microcontroller, they're relatively cheap, and they work with just about every language and controller around. However, they do have a few drawbacks, and one of them is their fairly limited colour range. There are 8 bits per colour (which give 255 levels), but we often limit our projects to only using the lowest of these numbers. For many projects, we only use the lowest 50 brightnesses in each colour, since going any brighter adds very little to the effect and drains power far quicker.

This problem is compounded by the fact that the lowest level (1) is already quite bright, so there's a big step between 0 and 1. Fortunately, we do have a trick up our sleeve that can improve things: Dithering. This is where we switch between two different brightness levels very quickly – the eye can't register the switch between brightness levels, so it views them as being at an intermediate brightness.

This is a very similar technique to Pulse Width Modulation (PWM) where we switch an LED on and off very quickly. The difference here is that we're not switching it on and off, but between different states that are already using PWM to adjust the brightness level. You could almost consider it PWM on top of

PWM. Let's take a look at a simple example that fades an LED in and out:

```
import time
import board
import neopixel

pixels = neopixel.NeoPixel(board.NEOPIXEL, 10,
auto_write=True)

dithertime = 0.01
numberoflevels = 10
while True:
    for j in range(numberoflevels):
        ditherproportion = j/numberoflevels
        for i in range(50):
            pixels[0] = (1,0,0)
            time.
sleep(dithertime*ditherproportion)
            pixels[0] = (0,0,0)
            time.sleep(dithertime*(1-
ditherproportion))

    for j in range(numberoflevels,1,-1):
        ditherproportion = j/numberoflevels
        for i in range(50):
            pixels[0] = (1,0,0)
            time.
sleep(dithertime*ditherproportion)
            pixels[0] = (0,0,0)
            time.sleep(dithertime*(1-
ditherproportion))
```

This code fades a NeoPixel for ten levels below 1 (which is typically the lowest level). You can use this same technique to add extra colour levels between any two adjacent levels.

This code is a little basic, but it shows the essential technique. In the main loop, there are two sub-loops – the first fades in and the second fades out. In each case, it loops through a set number of levels, and at each level calculates the amount of time the pixel should spend on or off, then loops through that 50 times before stepping up (or down) to the next level. The process is controlled by a number of variables:

- **`numberoflevels`** defines the number of increments between the 0 and 1 (or any other two levels) of brightness.
- **`dithertime`** defines the total time (in seconds) that is split between the two different levels of brightness. In theory, the smaller this is, the better effect you would get, but in practice, setting it too low introduced errors caused by the code taking longer to run than the sleep times, and the proportions in this naive calculation can get out of sync.

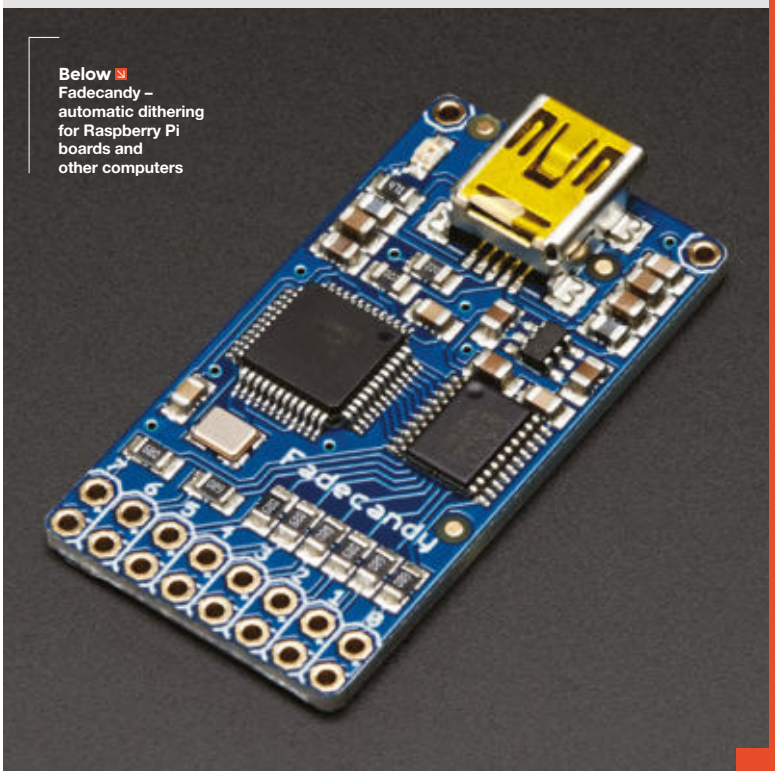
Dithering requires quite accurate timings, so the downside of this is that you won't really be able to do much other processing while this is running. However, you could use it to fade in or out. ◻

AN AUTOMATIC SOLUTION

Looking at it like this is a great way to get to understand how dithering works, but actually, it's not well-suited to running on your main microcontroller. The advantage of addressable LEDs, such as NeoPixels, is that you can just send data to them and let them handle the lighting. It's difficult to design the rest of your code in such a way as to let the microcontroller have enough time to also handle dithering.

One option here is to have a separate controller: use one microcontroller to run your code and another to handle the LEDs. While this might seem a little excessive, remember that there's a tiny microcontroller inside every NeoPixel anyway! If you've got a large LED installation, the cost of an extra microcontroller (especially as you don't need many features) may only work out to a small proportion of the total cost, so it can be a good value way of getting extra range on your LEDs.

You don't have to start from scratch. The Fadecandy is a microcontroller board specifically designed for dithering NeoPixels. The Fadecandy works over USB, so it needs a computer to drive it, not another microcontroller, but it works well with Raspberry Pi boards.



Below Fadecandy – automatic dithering for Raspberry Pi boards and other computers



Figure 1 Aluminium charge wells and a press-fitted bearing block made using boring techniques on the lathe and milling machine

Boring stuff!

Let's explore making holes larger using boring techniques on the lathe and the mill



Jo Hinchliffe

@concreted0g

Jo Hinchliffe is a constant tinkerer and is passionate about all things DIY space. He loves designing and scratch-building both model and high-power rockets, and releases the designs and components as open-source. He also has a shed full of lathes and milling machines and CNC kit!

Boring is the art of enlarging an already existing hole to a required size, with high accuracy and a good finish. Boring can also be used to create 'blind holes', where the hole does not go all the way through the workpiece and has a flat internal base, stepped holes with differing diameters, and tapered holes. In this article, we are going to look at two classic methods for boring holes using a lathe and a boring tool, and using a milling machine with a boring head.

The first boring job we want to create is some small aluminium flat-bottomed charge wells for a rocketry project, (**Figure 1**). They eventually will be bolted to a bulkhead and used to hold a small charge to split a rocket during flight, hopefully at the correct time! To begin, we chucked up some 21 mm diameter aluminium round bar in the lathe, faced the end, and centre drilled. Using a dead centre in the tail-stock to support the workpiece, we then turned the outside diameter down to the required size of 20 mm.

As the final design calls for a bolt hole in the bottom to connect these little pots to a bulkhead, it makes sense to create the starting hole for the bored

TOOLING UP



Boring tools come in numerous varieties. They can often be designed to work in either lathe tool holders or may also fit into boring heads. A boring head is a device that has a boring tool fitted and can (usually) be attached to a milling machine spindle. The boring head can adjust the diameter of the bore created incrementally by sliding the section holding the boring tool outwards. Our boring head takes ½ inch boring tools, and came supplied with a set of boring tools with carbide tips welded on, see **Figure 4** overleaf. Insert tools (where the carbide or other material tip is replaceable) are also available, such as the small boring bar in the centre of the above image. To the left of the image, we can see some high speed steel (HSS) boring tools; these tools are pieces of HSS which have been ground to create the geometry of a boring bar at their tip. The advantage of this is that a tool can be created with only a small amount of clearance required, so it can be used to start from and bore smaller holes. With the square shanks of the HSS boring bars, these tools are designed to be used in the lathe tool-post.



part at the size of the bolt hole of the final assembly – so we drilled a central hole using a 5.2 mm drill bit because we want to use an M5 fastener. Next, we fitted the small HSS boring tool to the lathe tool-post and ensured it was clamped well (**Figure 2**). We tried to leave the smallest amount of tool protruding from the tool-post to minimise the amount of deflection/movement of the tool. Boring, even in aluminium, and at this small scale, creates a lot of force as the tool tip is pulled into the inside of the workpiece. Next, we need to zero the tool onto the face of the workpiece and zero the very tip of the boring tool onto the internal wall of the 5.2 mm drilled hole. Having these zeroed means we can create the bore to an accurate depth (15 mm in our case) and an accurate internal diameter of 14 mm, leaving a 3 mm wall thickness.

Having bored the workpiece to the correct diameter, we then used the parting off tool to remove the workpiece from the stock. A parting off tool is a thin tool that cuts a neat, thin slot through the workpiece to eventually cut all the way through. It's a machining operation that requires some practice and patience, and is considered more difficult on smaller, lighter machines. It is relatively straightforward if the material is softer (like this aluminium) and if we are

> "Boring, even in aluminium, **and at this small scale, creates a lot of force**"

able to part off near the chuck, with less opportunity for the work to deflect away from the tool (**Figure 3**). We then finished these components on the milling machine, as they required some small slots milling into the walls for their intended use. You can see the finished items in **Figure 1.**

To perform boring operations on the milling machine, we are going to use a boring head. The boring head is a tool that is fitted to the milling machine spindle, and has numerous optional ways of mounting a boring bar in different positions (**Figure 4**). The boring head is similar in some ways to a compound slide on a lathe in that you can (with the aid of an Allen key) move the sliding section to which the boring bar is attached outwards and →

YOU'LL NEED

- **Milling machine or lathe**
- **Boring bars**
- **Boring head**
- **Drill bits**

Figure 2 Boring using an HSS boring bar in the lathe

Figure 3 Parting off on the lathe using a parting off tool to cut a small slot all the way through the workpiece



BACK ISSUES

Throughout this article, we are referring to lots of different parts of lathes and milling machines, and also a range of tasks that we have covered before in various articles. These include:

Lathe 101 issue 15
Milling 101 issue 14
Marking out issue 19
Drilling and tapping issue 20
Arbor press issue 24

These are all available for download for free on our back issues page at **hsmag.cc/issues.**

ON POINT



Many lathes are fitted with graduated wheels on the handles that allow the operator to dial in a particular amount, or to move a part of the machine to a known point. Some machines have fitted digital readouts and other devices to assist this. However, sometimes we want to repeat a movement accurately, and we can streamline this process by using a travel stop. Boring this flat-bottomed hole in this job is a good example of where a travel stop might be a useful accessory, as we want to make repeated passes to the same depth. A crude travel stop can be as simple as moving the carriage to the required stop position, and placing a clamp on the lathe ways that we can then repeatedly move the carriage up to. There are also numerous DIY carriage stop designs out in the wild. We made this 3D-printed and metal one for the mini lathe, for example.

QUICK TIP

Yes, we say it every article, but do remember to remove rings, stow away long hair and loose clothing, and wear your PPE before operating lathes or milling machines!

inwards, increasing or decreasing the cutting diameter the boring tool travels.

We wanted to drill a hole through a block of aluminium and then bore the top 8mm of the hole to a wider bore to accommodate a 30mm bearing. We wanted the bearing to be a 'press fit' where the hole would be bored slightly under-size, and then we use the arbor press to push the bearing in. Checking online for the required tolerance, we found that a hole that was under-size by 0.035mm was the correct fit



Figure 4
A boring head kit with a collection of different size boring bars

tolerance for this type of bearing. The accuracy of our machine means that the closest we could get is 1/100th of a millimetre, not a 1/1000th, so we aimed for 0.04mm under-size, making the target diameter 29.96mm. We began by marking out and centre-punching the position in the block we wanted the centre on, and then clamped the workpiece to the milling machine table using some standoff blocks underneath to allow us to drill through the piece.

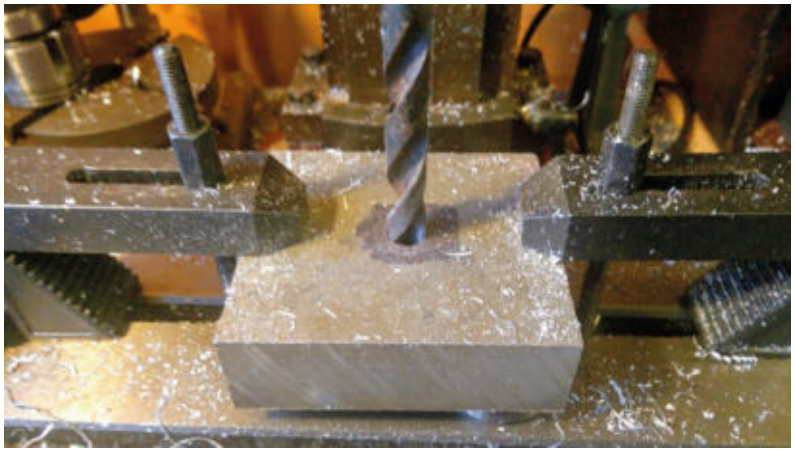
The hole through the block isn't critical for the design, but needed to allow a 10mm shaft to travel through it with clearance. The boring bar we used needed around a 9mm starting hole, so we used numerous increasing sized drills (6mm, 8mm, 9.5mm) to eventually drill the final hole at 10.5mm (**Figure 5**). Similar to boring on a lathe, the boring bar fitted to a boring head has a lot of force applied to it during cutting and can be made to deflect, so if there is a choice, using the shortest and widest/strongest boring bar is always the preferable option.

> "The boring bar fitted to a boring head **has a lot of force applied to it during cutting**"

LET'S GET OUR HEAD STRAIGHT

Opinions differ on how you should set the boring bar tool in the boring head with regard to the angle of the tip of the tool with the workpiece. Many say that the tip of the boring tool should be slightly behind an imaginary centreline from the centre of the spindle to the work surface, or should be set at a 'neutral rake', which means it is exactly perpendicular. In our experiments, slightly setting the tool a few degrees behind perpendicular seemed to get better results.

Having set the tool in the boring head, and ensured that the boring bar retaining grub screw was tightened, we then loosened the grub screw that locks the sliding ways. We brought the tool outwards, and then lowered the spindle to touch the boring tool tip onto the surface of the block using a feeler gauge. We then zeroed the Z-axis dial on the milling machine. We recentred the tool over the drilled hole, and used the fine feed Z-axis dial to lower the tool 8mm. We then set the travel stop on the Z-axis column so that we could repeatedly bring the tool down to that depth (the depth of the bored hole was a much less critical dimension than the diameter, so this method was accurate enough). Finally, before commencing, we



SLOW GOING

Typically, small boring heads in smaller milling machines are run at slower rpm than an equivalent diameter drill bit. Searching online, you can find lots of different advice on speeds and feeds, and results will vary between machines and, obviously, materials. We began our cutting at around 350 rpm as we found this a good speed that gave reasonable results in aluminium. It's worth remembering though that boring to increase a hole diameter will mean that, while the rpm will remain the same, the relative cutting speed at the tip of the tool will increase as the diameter of the cut increases. Remembering to reduce rpm to keep a consistent cutting speed is important to maintain a similar finish.

slid the boring head outwards to touch the tip of the boring tool to the inside of the hole. We checked to see what position the graduated dial on the boring head was in so we could move the boring head a known amount to create a particular size of cut.

We dialled in a first cut bringing the boring bar out to add 0.8 mm to the diameter (so a 0.4 mm cut). Making sure to lock the sliding part of the boring head with the centre grub screw on the sliding section, we fed the boring bar down steadily until it reached the required depth at the depth stop. We repeated this time after time, bringing the hole closer to the desired 29.96 mm. Instead of relying on adding up the amount we cut in each pass, we performed running checks to see what size the hole was at a given point. We can lift the boring head and spindle clear of the workpiece and then use telescopic gauges to measure the bore. A telescopic gauge has two sprung sections in the tip that can be compressed inwards – squeezing them together we can insert them into the bore. The most accurate way to use them is to place them at an angle in the hole and lock them by tightening the screw at the base of the handle, then forcing them into a straight position across the bore, pushing the telescoping pistons into a locked position matching the hole diameter (**Figure 6**). Remove them, and then accurately measure the width of the gauge with a micrometer (**Figure 7**). With some practice, this is can be an extremely accurate method for measuring internal diameters, second only to an internal bore micrometer or a more high-end bore gauge. If these tools aren't available, a less accurate way is to use the internal jaws on a set of vernier callipers, but care must be taken to try and ensure that you measure the diameter at the widest point.





Continuing to make cuts, we brought the hole to the required diameter, and we also did a further small bore to allow the inner race of the bearing to spin freely. Finally, we used the arbor press to push the bearing snugly into the block, which is a very satisfying thing to do (**Figure 1**). It's useful to have the capability to bore, and although the purchase of boring tools, particularly a boring head, might seem costly, the fact that you have a huge range of diameters you can cut as through, blind, or stepped holes is a great benefit to many projects. □

Figure 5
The block of aluminium having been marked out and drilled to create the 10.5 mm through-hole

Figure 6
About halfway through the job, checking diameter progress with a telescoping gauge

Figure 7
Using a micrometer to measure the width of the telescoping gauges to discover the bore diameter

Maker's toolbox: Arc welders

Molten metal joints

TOOLS Provided by SCREWFIX



Above The knob turns the current up and down – that's the only control you have with an AC stick welder

Welding is the art of melting two or more metals together so that they make one continuous bit of metal (OK, technically, they don't have to be metals, but for most purposes they are). Unlike, say, soldering where you melt one metal and use this molten metal to join other bits of non-molten metal together.

While you can weld a range of metals, for most hobbyists steel is the only practical metal for welding. It's cheap, easily available in a range of shapes, and can be worked with simple tools.

There are a few ways of welding steel, and they can be split depending on the way they heat the steel. Some use gas flames, some use electric sparks, and some use other exotic methods (such as friction welding). We'll look at electric welding here because it's the most accessible method for most hobbyists as it doesn't (necessarily) require gases, which can be both hard to store and transport.

There are three basic ways of welding with an electric spark:

- Shielded Metal Arc Welding (commonly known as stick) is the simplest. You have a welding unit which is basically a big power supply that can create a very high current. Onto this, you attach a stick-shaped electrode which is covered in flux powder. The electrode melts as you weld, to provide 'filler' material that goes into the joint to make sure that there's enough molten metal to make a strong joint. It's a bit messy, it's the least 'neat' of the electric methods, but the equipment is cheap, and it can produce decent welds (though struggles on thin sheets).

- Metal Inert Gas (MIG) welding can work in two ways: gasless and with gas. Without gas, it uses electrodes with flux, much like stick

welding does, except that this comes on a roll of wire and is fed out as you weld, which allows more control than with stick welding. With gas, the electrode doesn't contain a flux, but an inert gas is blown over the weld to protect it as you go. This is neater as there's no flux burning in the weld.

- Tungsten Inert Gas (TIG) allows the neatest joints. In both stick and MIG welding, the electrode is also the filler metal. In TIG, there's a tungsten electrode that doesn't melt as you weld, so the spark just provides a source of heat. You then use a separate rod of filler metal that you can add as necessary. This separation of heat and filler metal gives you more control over the process, and TIG welding can produce neater joints on thinner metal than the other two methods.

Which method you opt to learn depends on what you want to achieve. Here, we're going to look at stick welding because that's the most accessible – the equipment's affordable and, while we recommend getting instruction from a skilled welder if possible, you can learn the basics from books and online videos. Do pay attention to safety – be aware of the risk of fire, eye damage, burns, and fume inhalation.

PICKING A WELDER

There are a few features to look out for in welding power supplies. The most basic is what types of current it supports. You can weld with AC power, and DC (which can be either way around – known as DC Electrode Positive or DC Electrode Negative). Many cheap stick welders only have AC mode, as that's far simpler electrically. AC welding has a bit more splatter and doesn't have the penetrating power of DC welding. However, it's perfectly possible to get good results with AC welding.

Welding requires very high currents – up to, and beyond, a hundred amps. How much current you need depends on the electrode you're using and what you're welding. The thicker the electrode (which generally means the thicker the material you're welding), the more current you need.

The final thing that you may want to look for in a stick welder power supply is the duty cycle. Producing such large currents will heat up the electrics in the welder, and when they get too hot, the welder will cut out. The duty cycle is a measure of how long you can expect to use the welder without it overheating. It's often given as a percentage at a particular current.



A 10% duty cycle at 100A means that you can use the welder 10% of the time if you're welding at 100A.

The welder we are testing is an Impax IM-ARC140. The maximum current is 105A (the '140' in the name is the model number, and unrelated to welding current), which it can sustain at 10% duty cycle. It is fan-cooled (some ultra-budget units are passively cooled) and AC only. In welding terms, both in price (£84.99) and features, this is entry-level. However, that doesn't mean that you can't do useful welding with it.

> How much current you need depends on the **electrode you're using and what you're welding**

105A is more than enough current for many basic welding tasks (you need more current for thicker electrodes and welding thicker materials). We didn't have any trouble welding 5mm steel with this, with some headroom, but if you're planning on joining significantly thicker bits of steel than this, you might need a more powerful machine.

AC welding doesn't have the best reputation – in a perfect world where we could all have whatever equipment we wanted, then we would recommend a DC machine (and probably a MIG or TIG machine) – but if you're just starting out, or not looking to do work on areas with fine detail, AC stick welding with a machine such as this is a good option. It's easy to learn, relatively cheap to kit ourself out, and great fun to do.



Ben Everard

@ben_everard

Ben loves cutting stuff, any stuff. There's no longer a shelf to store these tools on (it's now two shelves), and the door's in danger.

Above
Stick welding equipment is fairly straightforward – it's just the power supply and two electrodes

TUTORIAL

Make some noise with Raspberry Pi

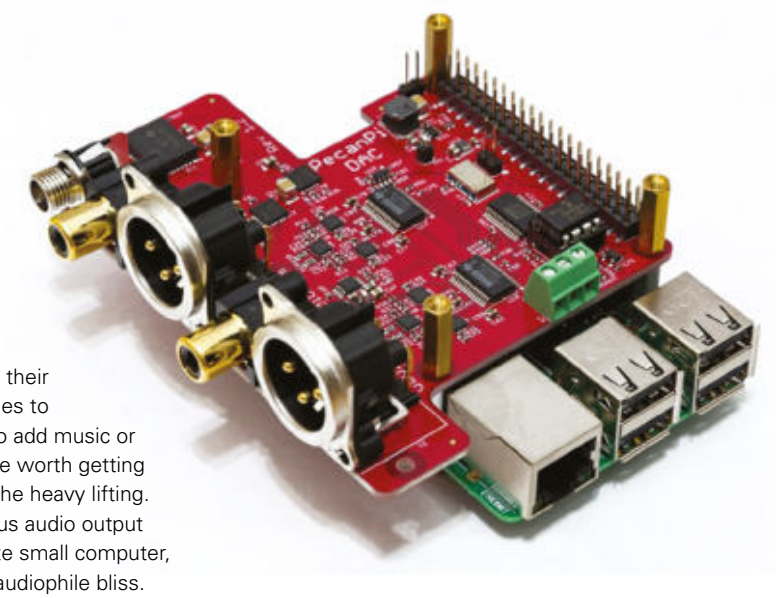
If your amazing project is a little too quiet, add high fidelity sound with Raspberry Pi



PJ Evans

@mrpjevans

PJ Evans is a developer and wrangler of the Milton Keynes Raspberry Jam. He runs a LoRa gateway, which is probably the nearest he'll get to his own radio breakfast show.



Above The PecanPi HAT features best-in-class components and dual DACs for superior audio reproduction

It's no surprise that we love microcontrollers here at HackSpace magazine. Their versatility and simplicity make them a must for electronics projects. Although a dab hand at reading sensors or illuminating LEDs, Arduinos and their friends do struggle when it comes to high-quality audio. If you need to add music or speech to your project, it may be worth getting a Raspberry Pi computer to do the heavy lifting. We're going to look at the various audio output options available for our favourite small computer, from a simple buzz, through to audiophile bliss.

GET BUZZING

The simplest place to start is with the humble buzzer. A cheap active buzzer can be quickly added to Raspberry Pi's GPIO. It's surprisingly easy too. Try connecting a buzzer's red wire (positive) to GPIO pin 22 (Broadcom numbering) and the black wire (ground) to any GND pin. Now, install the GPIO Zero Python library by typing this at the command line:

```
sudo apt install python3-gpiozero
```

Create a file called **buzz.py** in your favourite editor and enter the following:

```
import time
from gpiozero import Buzzer
buzzer = Buzzer(22)
buzzer.on()
time.sleep(1)
buzzer.off()
```

Run it at the command line:

```
python3 buzz.py
```

You should hear a one-second buzz. See if you can make Morse code sounds by changing the duration of the `sleep` statement.

PASSIVE BUT NOT AGGRESSIVE

Raspberry Pi computers, with the exception of the Zero range, all have audio output on board. The original Raspberry Pi featured a stereo 3.5 mm socket, and all A and B models since feature a four-pole socket that also includes composite video. This provides your cheapest route to getting audio from your Raspberry Pi computer. A low-cost passive speaker can be directly plugged in to provide sound, albeit probably quieter than you'd like. Of course, add an amplifier or active speaker and you have sound as loud as you like. This is the most direct way of adding sound to your project, but how to get the sound out?

Normally, the Raspbian operating system will recognise that an audio device has been connected and route audio through it. Sometimes, especially if you've connected an HDMI monitor with sound capability (e.g. an HDMI TV), sound will not come

YOU'LL NEED

- **Raspberry Pi 4**
- **Audio device(s) as featured**

out of the correct device. To fix this, open up a terminal window and run **sudo raspi-config**. When the menu appears, go to 'Advanced Options' and select 'Audio', then select the option to force the output through the audio jack. You may need to reboot Raspbian for all changes to take effect.

PLUG AND PLAYBACK

A USB sound device is another simple choice for audio playback on Raspberry Pi. Literally hundreds are available, and a basic input/output device with better audio quality than the on-board system can be purchased for a few pounds online. Installation tends to be no more complicated than plugging the device into the USB port. You may need to select the new output, as the underlying audio system, ALSA (see box below), may mute it by default. To fix this, run **alsamixer** from the command line, press **F6** to select the new sound device, and if you see 'MM' at the bottom of the volume indicator, press **M** to unmute and adjust the volume with the cursor keys.

> As they are low-power, USB devices **do not tend to feature amplification**

Unsurprisingly, when choosing your USB sound device, you can start at a few pounds and go right up to professional equipment costing hundreds. As they are low-power, USB devices do not tend to feature amplification, unless they have a separate power source.

LET'S PLAY

The simplest way to play audio on Raspbian is to use OMXPlayer. This is a dedicated hardware-accelerated command-line tool that takes full advantage of Raspberry Pi's capabilities. It sends audio to the analogue audio jack by default, so playing back an MP3 file is as simple as:

```
omxplayer /path/to/audio/file.wav
```

There are many command-line options that allow you to control how the audio is played. Want the audio to loop forever? Just add **--loop** to the command. You'll notice that when it's running, OMXPlayer provides a user interface of sorts, allowing you to control playback from within the terminal. If you'd just like it to run in the background without user input, run the command like this:

```
omxplayer --no-keys example.wav &
```

Here, **--no-keys** removes the interface, and the ampersand (**&**) tells the operating system to run the job 'in the background' so that it won't block anything else you want to do.

OMXPlayer is a great choice for Raspbian, but other players such as mpg321 are available, so find the tool that's best for you.

Another useful utility is speaker-test. This can produce white noise or vocal confirmation so you →

ALSA AND PULSEAUDIO

There's often confusion between these two systems. Raspbian comes preinstalled with ALSA (Advanced Linux Sound Architecture), which is the low-level software that makes sound work. It comes with a range of utilities to control output device, volume, and more. PulseAudio is a software layer that sits on top of ALSA to provide more features, including streaming capabilities. Chances are, if you need to do something a bit more clever than just play audio, you'll need to install a PulseAudio server.



QUICK TIP

Need to adjust the volume from the command line? Just enter **alsamixer** to access a simple control screen.

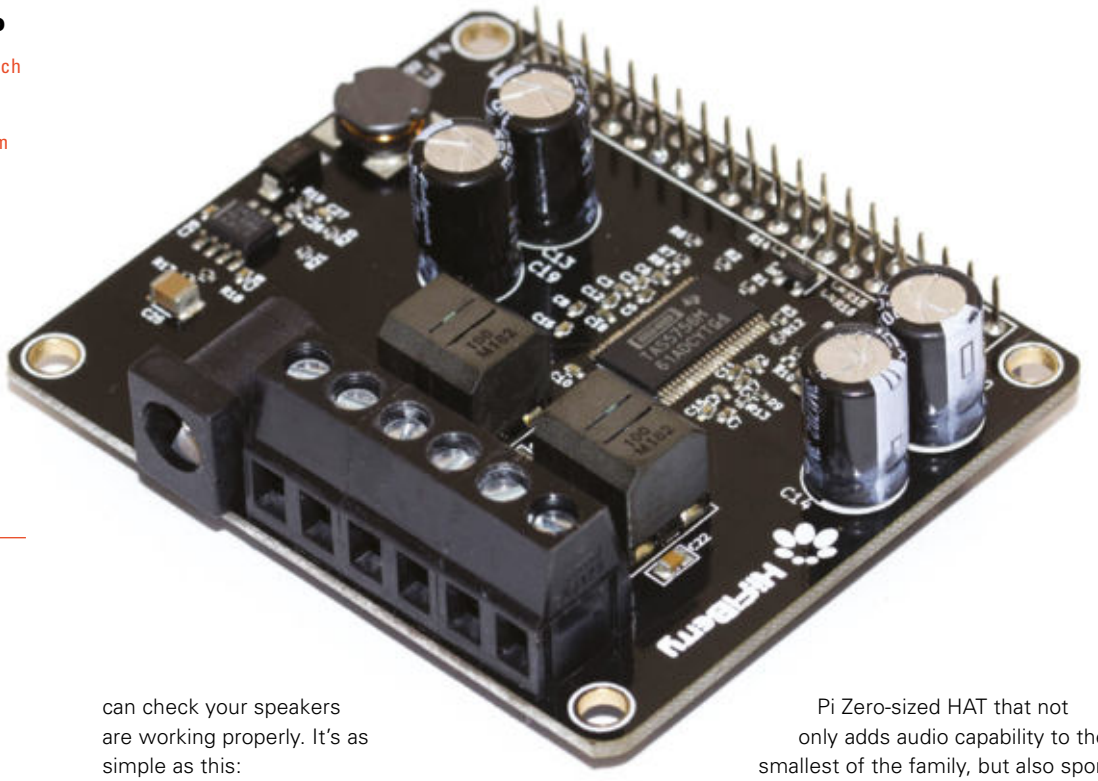


Above
Small, cheap, and fun, the Speaker pHat features a 3W speaker and LED VU meter

Below
Need a simple solution? USB audio devices come in all shapes and sizes but are mostly plug-and-play

QUICK TIP

Free software such as SnapCast can allow Raspbian devices to stream and play back audio in sync.



can check your speakers are working properly. It's as simple as this:

```
speaker-test -t wav -c 2
```

The first parameter sets the sound to be a voice, and the **-c** tests stereo channels only: front left and front right.

PHAT BEATS

If space is an issue, a Raspberry Pi 4, amplifier, and speaker may not be what you have in mind. After all, your cool wearable project is going to be problematic if you're trailing an amplifier on a cart with a 50-metre extension lead powering everything. Luckily, the clever people at Pimoroni have you covered. The Speaker pHAT is a Raspberry Pi Zero-sized HAT that not only adds audio capability to the smallest of the family, but also sports a 3W speaker. Now you can play any audio with a tiny device and a USB battery pack. The installation process is fully automated, so no messing around with drivers and config files. Once the script has completed, you can run any audio tool as before, and the sound will be routed through the speaker. No, the maximum volume won't be troubling any heavy metal concerts, but you can't knock the convenience and form factor.

PLAYING THE BLUES

An easy way to get superior audio quality using a Raspberry Pi computer is Bluetooth. Recent models such as the 3B, 4, and even the Zero W support Bluetooth devices, and can be paired with most Bluetooth speakers, even from the command line. Once connected, you have a range of options on size and output power, plus the advantage of wireless connectivity.

Setting up a Bluetooth connection, especially if you are using the command line, can be a little challenging (see the 'Bluetooth cheat sheet' box on the opposite page). There is a succinct guide here: **hsmag.cc/N6p2lB**. If you are using Raspbian Desktop, it's a lot easier. Simply click on the Bluetooth logo on the top-right, and follow the instructions to pair your device.

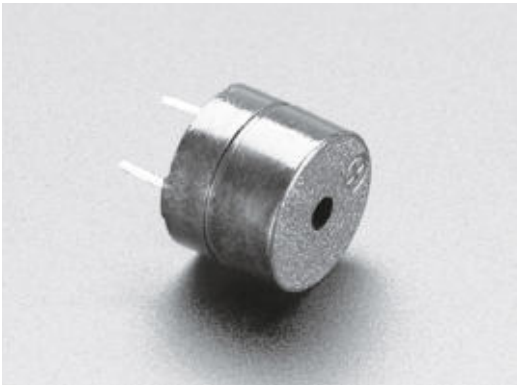
Above
Many DACs also come with on-board amplifiers. Perfect for passive speakers

Below
Money no object? The Allo Katana is a monster DAC, and weighs in at £240, but outperforms £1000 equivalents



FILE TYPES

There are command-line players available for just about every audio format in common use. Generally, MP3 provides the best balance of quality and space, but lower bit-rates result in lower sound quality. WAV is completely uncompressed, but can eat up your SSD card. If you don't want to compromise on audio quality, try FLAC, which is identical in quality to WAV, but much smaller. To convert between audio types, consider installing FFmpeg, a powerful audio and video processing tool.



Above
Need to keep it simple and under a pound? A simple active buzzer is what you need

If you find OMXPlayer isn't outputting any audio, try installing mpg321:

```
sudo apt install mpg321
```

And try again:

```
mpg321 /path/to/audio/file.mp3
```

BUT SERIOUSLY

If your project needs good audio, and the standard 3.5mm output just isn't cutting it, then it's time to look at the wide range of DACs (digital to analogue converters) available in HAT format. It's a crowded market, and the prices vary significantly depending on what you want from your device. Let's start at the lower end, with major player HiFiBerry's DAC+ Zero. This tiny HAT adds 192kHz/24-bit playback via two RCA phono ports for £12.50. If you're serious about your audio, then you can consider the firm's full HAT format high-resolution DAC+ Pro for £36, or really go for it with the DSP (digital sound processing) version for £67. All of these will require amplification, but the sound quality will rival audio components of a much higher price.

> "It's a crowded market, and the prices vary **significantly depending on what you want**"

If money is no object and your project requires the best possible reproduction, then you can consider going full audiophile. There are some amazing high-end HATs out there, but one of the best-performing ones we've seen is the PecanPi DAC. Its creator Leonid Ayzenshtat sourced each individual component carefully, always choosing the best-in-class. He even used a separate DAC for each audio channel. The resulting board may make your wallet wince at around £200 for the bare board, but the resulting audio is good enough to be used in professional recording studios. If you've restored a gorgeous old radio back to showroom condition, you could do a lot worse than add the board in with a great amp and speaker.

BLUETOOTH CHEAT SHEET

If you want to pair a Bluetooth audio device (A2DP) on the command line, it can be a little hairy. Here's a quick guide:

First-time installation:

```
sudo apt-get install pulseaudio pulseaudio-module-bluetooth
sudo usermod -G bluetooth -a pi
sudo reboot
```

Start the PulseAudio server:

```
pulseaudio --start
```

Run the Bluetooth utility:

```
bluetoothctl
```

Put your speaker into pairing mode. Now, within the utility, run the following commands (pressing **ENTER** after each one):

```
power on
agent on
scan on
```

Now wait for the list to populate. When you see your device…

```
pair <dev>
```

Where **<dev>** is the displayed long identifier for your device. You can just type in the first few characters and press **TAB** to auto-complete. Do the same for the following steps.

```
trust <dev>
connect <dev>
```

Wait for the confirmation, then enter:

```
quit
```

Now try to play some audio using aplay (for WAV files) or mpg321 (for mp3). These instructions are adapted from the guide by Actuino at **hsmag.cc/N6p2IB**.

How to maximise the organisation of your workshop

Improving efficiency in your shop through better organisation and smarter storage



Gareth Branwyn

@garethb2

Gareth has been a lifelong practitioner (and chronicler) of DIY tech, media, and culture. He is the author of ten books, including *Tips and Tales from the Workshop*, and is a former editor for Boing Boing, Wired, and Make:.

A **few years ago, this author finally broke down and reorganised his workshop.** Not to overstate things, but it significantly changed the quality of his work life. The reorganisation process wasn't fun – it was a lot of work, and it wasn't that cheap, but it offered a workshop reboot that he thanks himself for every day now. Here are some ideas, organisational tools, and storage technologies to consider when planning your next shop reorganisation.

First-order retrievability This is a concept that Adam Savage of **Tested.com** is big on. Plan out your shop so that your everyday tools are closest to you and least used tools are farthest away.

Everything on quality, lockable casters Of course, tool needs and priorities change, so to be able to reconfigure your tools as your daily work needs change, put all tables, storage cabinets, supply carts, etc. on lockable casters. And don't be tempted to get cheap casters. Buy the highest quality ones you can afford.

Arrange tools to minimise dead space When designing your shop layout, think about your workflow and how you physically approach and use your machines. Group machines so that their dead spaces (the sides you don't interact with) are butted up against each other.

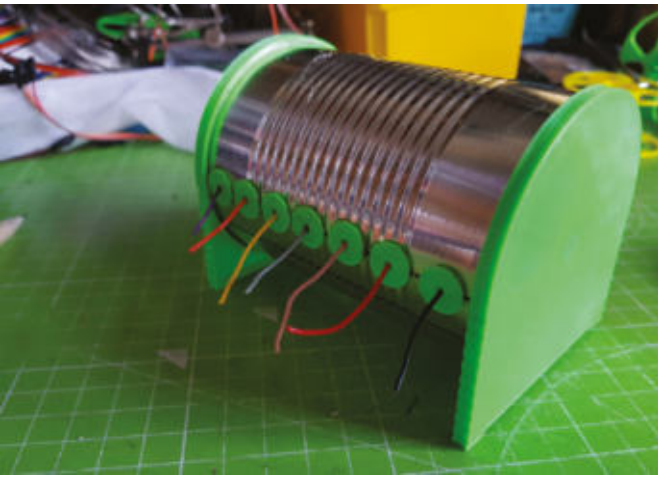
Use a space layout program (or paper layouts) There are numerous, free space planning tools available online to lay everything out and be able to move everything around. I like the free Java-based app at tool supplier **Grizzly.com**. You can also use SketchUp. Or just graph paper and cut-outs of your main shop objects.



Above
My LED panel light, based on a project by Adam Savage on the Tested YouTube channel (search for 'Custom Workbench LED Lamp')

Use consistent storage systems It is really helpful and efficient to use multiples of the same storage cases, bins, buckets, shelving units, etc. If you settle for a rag-tag collection of odd boxes, bins, and recycled jars and cans, it will only lead to wasted space, and it will come to frustrate you.

Overlight the space You need light. Lots of light. More light than you think. Thankfully, good shop lighting can be very affordable these days. High intensity, linkable LED utility shop lights can be found online for under £100. For desk lighting, this



author uses LED panel lights (created for video recorders) attached to goose-necks.

Tool redundancy For cheap tools that you need really often (rulers, scissors, utility knives, markers, hammers, drivers, etc.), buy multiples of them and keep them all over the shop. Make special holders for them so that they are always where you know you can find them.

Groups tools and supplies by activity, not type It's easy to think that the way to store tools and supplies is to group them all together (glues in this bin, screwdrivers in this case, rags in this bucket), but for commonly done tasks (let's say paint-priming), try storing everything you need (spray-paint cans, face mask, latex gloves, masking tape) in a single, clearly labelled storage bin.

STORAGE IDEAS

Tin can wire organiser HackSpace magazine contributor Andrew Lewis has created a set of 3D files for printing the parts for converting a food can into a spool-based wire dispenser. The resulting unit holds seven spools of wire. The CAD files can easily be resized for different-sized cans. You can find the files at **hsmag.cc/GgyGk9**.

Storage cases with removable bins If lots of your components, hardware, and other shop bits and bobs are small, consider investing in plastic portable storage cases with removable bins inside. You can find these at online tool warehouses for around £6–10/each. This author bought several dozen during my shop reorg and still has some spares

Storage case racks (on wheels) It's common for those who use the above storage cases as a central part of their storage tech to build simple wooden carts on casters to house them. Having a number of these with parts and hardware well-organised in the cases is a fantastic way of pulling in the components you need when you need them (and then pushing them out of the way when you're done).

Use see-through plastic bins Plastic bins with lids are another great, and relatively inexpensive, storage solution. Look for craft and home stores to put these on sale frequently at deeply discounted prices. If you find them, get the clear, rather than milk-white or coloured bins. This way, you can see the contents in each bin at a glance.

Shop nail salon tech It may surprise you, but a great place to find storage solutions for your shop is in the nail salon 'aisle' of your favourite online mega-retailer. Nail polish racks, shelving, polish mixing machines, and even nail art paintbrushes are all useful in your shop. And most of the products are surprisingly cheap. This author spent US$40 on a laser-cut storage rack for hobby paints, and it took an hour to assemble it. He also got a clear acrylic nail polish rack for half that price, it took about five minutes to assemble, and it holds more paints!

Pound shop/dollar store storage tech Don't forget to look for storage solutions at your local pound shop/dollar store. Zippered pencil cases are great for storing things like sorted data cables. Trading card sleeves can be organised in binders to sort electronic components and other small items. And they have all sorts of other bins and containers you can press into service in the shop.

Use the walls and even the ceiling If you have rafters in your shop space, consider using brackets to hold materials in the spaces between the rafters. Or, you can build (or buy) pull-down shelving that folds up into the rafters like attic stairs. You should also think about your wall space (and even the backs of doors) and how you might best utilise them. Go vertical! Work the cube! ◻





Above
There are dozens of simple plans for building these storage case cabinets online. This one was built by Andrew Lewis

Left
Store your wire and keep waste out of landfill

Below
South African YouTuber, Jean of 'Woodshop Junkies', has a video on his channel about building in-ceiling shelving

QUICK TIP

Subscribe to Facebook groups and YouTube channels that encourage shop organisation and clever storage solutions. I follow the 'Shop Hacks' group on Facebook, and frequently get inspired by what members are sharing there.

TUTORIAL



Absolutely Belting: Efficient drive systems

Drive your projects more effectively by choosing the proper belt



Dr Andrew Lewis

@monkeysailor

Dr Andrew Lewis is the owner of **Shedlandia.com**, a restorer of old tools, a fabricator for hire, a research scientist, and a founder member of the Guild of Makers.

Above
Different belts are suited to different tasks – it can be difficult to pick the right one

Most people have never really thought about drive-belts, yet they're one of the most common types of power transfer used in machinery. Belts are compact, easy to handle, and they have lots of variable properties that affect the way a machine will run. They're also more complicated than you might expect. In this article, you'll see how to choose and use the best type of belt for the project you have at hand, and learn some of the terminology used to describe a belt.

> If you have a pillar drill, then you'll probably find that **there's a pulley-driven system inside**

If you have a pillar drill, then you'll probably find that there's a pulley-driven system inside. Most pillar drills use arrangements of pulleys and V-belts to change the speed of the shaft that drives the chuck, and in some workshops, those belts and pulleys might have been left in place for decades. That's a problem, because the materials used to make belts might be long-lasting, but they're ultimately perishable and will wear out with use. Old belts can stretch, rip, lose their flexibility, and eventually disintegrate. A machine like a pillar drill with an old belt will still work, but there will be less power transferred to the chuck, the reduced flexibility of the belt will make the power transfer uneven, and there will be increased noise and vibration coming from the motor.

BELTS ARE COMPLICATED

A V-belt and pulley work together because of friction, and the arrangement is actually referred to as a friction drive system in some literature. The friction between the surface of the belt and the pulley means that when the pulley turns, the belt gets dragged along for the ride. At the opposite end of the drive chain, friction makes the pulley turn as the belt is dragged around it. V-belts are more



popular than flat belts because the V shape increases the amount of belt surface in touch with the pulley. More surface area means more friction, and more friction means more power transfer. That's why the tension on a drive-belt is important – less tension means less friction, and more chance that a belt will slip. Pretty much every belt-driven system will have some mechanism to hold the belts in tension. This might be done by the weight of the motor hanging on a vertical belt or by a sprung pulley, called an idler pulley, that pushes against a belt to increase surface contact with the other pulleys and reduce slack.

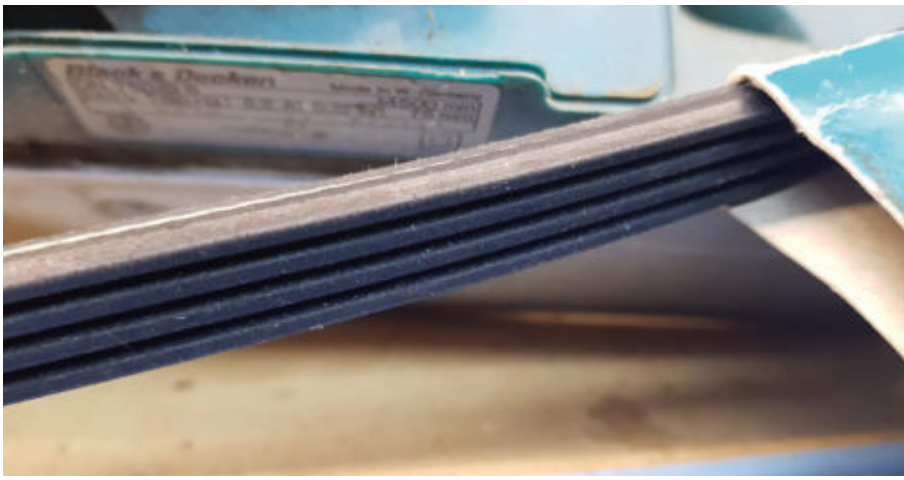
BUYING A BELT

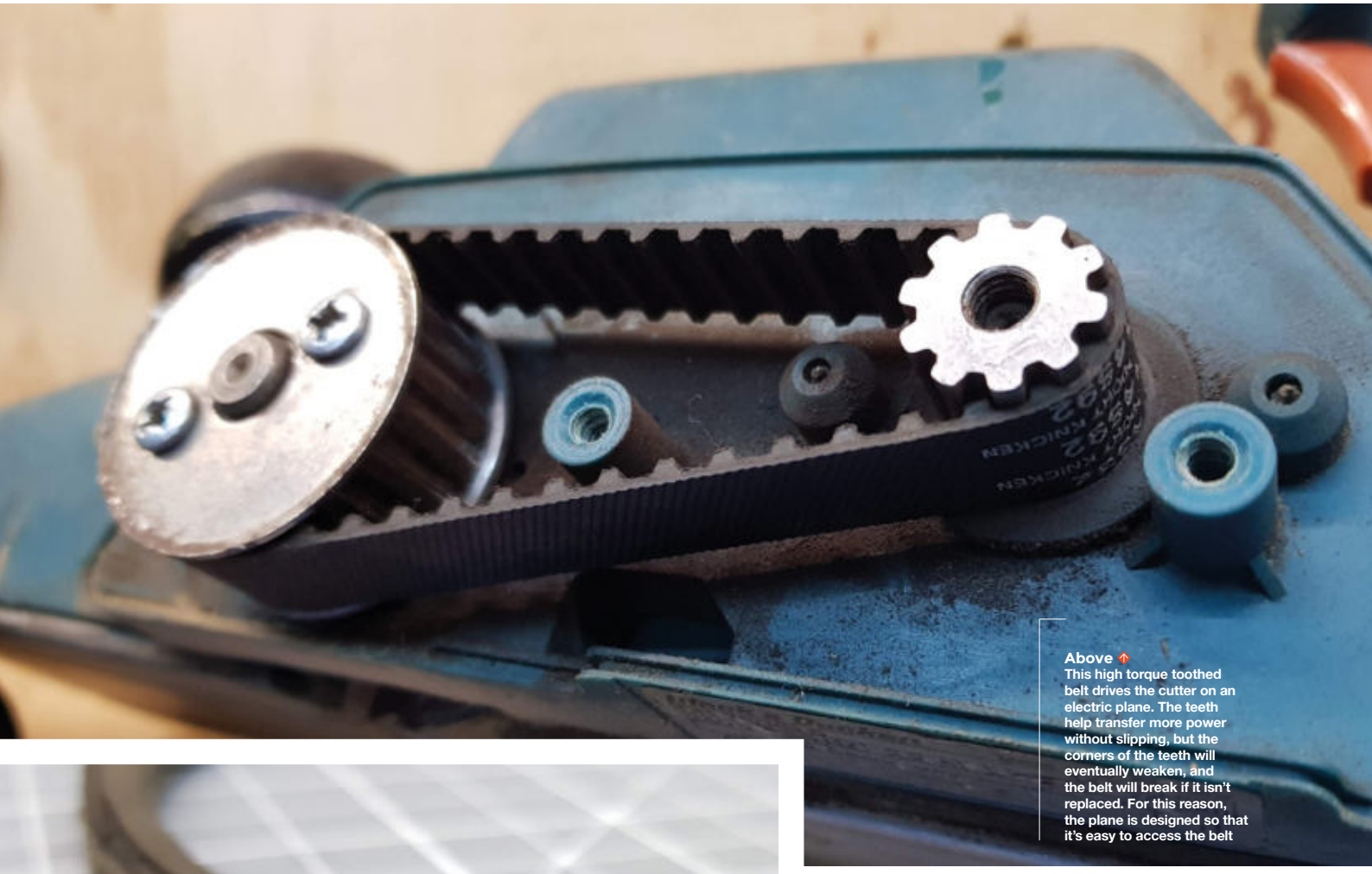
Ordering the right size of belt from a supplier is not that hard if you understand how the numbering system works. For V-belts, the original coding system was developed in the early 1900s, and referred simply to the profile size and shape (ranging from A–E), and then a length given in inches. Using this system, an A24 belt would have profile A, and have a total length of 24 inches. As the years went by, the system was extended and changed to include smaller and larger sizes, metric belt sizes, and individual manufacturers started to come up with their own codes to specify unique features like chemical and thermal resistance, improved flexibility, and resistance to shock loading. It's quite easy to find profile charts online and look at the different manufacturers' specifications for belts, but the basic starting point for classic V-belts is that profile sizes increase in the order M, Z, A, B, C, D, E, with E being the largest. These classic-type belts are wedge-shaped, and have an angle of about 36 degrees. Narrow profile belts are numbered 3V, 5V, or 8V and have a steeper wedge angle than a standard V-belt. This means they can handle a much greater load without slipping. Other common types include metric belts, which are essentially metric equivalents of the original belt series, prefixed with SP, meaning that an SPC belt will have approximately the same width as a classic C belt, but with a deeper wedge profile.

If you have a more modern precision drill-press or a lathe with a motor speed controller, then you might find that the cone gears and V-belts have been replaced with a single toothed belt (also sometimes called a timing belt), which is designed to fit onto a toothed pulley. The advantage of a toothed belt in a system like this is that it doesn't just rely on friction to transfer the power of the motor, and so it can transfer all of the motor power with a much smaller belt. However, this sort of robust configuration isn't always a good idea. With a friction-driven belt system, an excess load will make the belt slip, almost like a clutch. This can protect the motor from being overloaded, and can also protect other parts of a machine from being damaged. With a toothed belt, slipping is much less likely, so an overloaded belt will either strip its teeth, or some other part of the machine will get bent or burnt out. In cases like this, a V-belt can be a safety feature, designed to slip or fail so that other more expensive parts of the machine don't get damaged. →

Top Left
It's not unusual to see pillar drills with two or three cone pulleys like this to control the speed of the drill. It's not the most convenient method of speed control, but it is cheap and effective

Below
If you need to drive a heavy or transient load (the sort of load that wants to snatch and snag), you might need to use more than one V-belt joined in parallel, and choose a belt profile with a steeper angle to increase the contact between pulley and belt





Above
This high torque toothed belt drives the cutter on an electric plane. The teeth help transfer more power without slipping, but the corners of the teeth will eventually weaken, and the belt will break if it isn't replaced. For this reason, the plane is designed so that it's easy to access the belt



Above
The edge of this belt has worn, and parts of the outer have started to peel away. On the inside of the belt, the surface has started to become very shiny where it has slipped on the pulley, and you can see a section where the inner metal banding is exposed

WHAT CAN GO WRONG? WELL...

A V-belt is usually made from a wedge shape of rubber material with an internal metal (or plastic) band that resists stretching, and a flexible rubber or plastic backing. Any of these parts can fail and make a piece of machinery work poorly. One of the most common faults in a belt is that it has lost its flexibility. You can test this very easily by holding the belt and looking at it. If the belt holds the shape of the pulleys and doesn't spring back into a round, then it's not flexible and will vibrate.

Another common issue that you will find with all types of belt is stretching and general wear. The metal band around a belt can stretch over time, and that will make the machine lose power through slippage. Increased slippage can also make the belt overheat, wear away the compressing surface of the belt, and in extreme cases cause a belt to smoke or catch fire. So it's worth checking that the overall length of the belt is within tolerance for the machine, and that any idler pulleys are actually doing their jobs.





WITH TEETH

Finding the right toothed belt is not quite as straightforward as finding a plain V-belt. If you're familiar with 3D printers, you've probably heard of and been confused by GT, 2GT, and GT2 belts. You might even have heard of HTD belt profiles. Toothed belts (also known as timing belts or synchronous belts) fall into the broad categories of trapezoidal and curvilinear. Trapezoidal belts are good for accurate positioning with minimal backlash, and they're a traditional favourite in automation systems. The problem with these types of belt is that there's some evidence that the sharp angles on the tooth wear out more quickly under strain. Curvilinear belts are a modern response to this problem, and have a more rounded profile. Curvilinear belts are sometimes called HTD (high torque drive) belts, after the designation given to them by their developer. The common GT2 (Gates Tooth 2) belts found in a 3D printer are also curvilinear belts, and this is where things start to get complicated. Different manufacturers have different brand names for their own products, which are all slightly different. The only real way to work with these numbers is to search online, pick a suitable type for your application, and stick with it.

You should be able to detect problems like overheating, chemical damage, and physical damage just by looking carefully at the belt. If you can see damage, cracks, or exposed metal bands on a belt, you should replace it immediately. A damaged belt could snap at any time and cause significant damage to people or machinery. When you are replacing a belt, consider the cause of the damage and get an appropriate replacement. Using the wrong type of belt in a harsh environment is sure to cause difficulties and disappointment.

If you need to get a belt to wrap around a small pulley, a standard V-belt might not be flexible enough to do the job. You can use a notched or cogged belt (which is not the same as a toothed belt) to get around this problem. A cogged belt has part of the internal surface notched away like an infinite Toblerone, so it can be bent more tightly around a pulley at the expense of some surface area.

Once you've identified and replaced a faulty belt, one final tip is to record the belt information somewhere on the machine. Make sure that you include the model number of the belt and the date that it was changed. That way, you'll be able to order a replacement in future without struggling to read the worn-out letters on a broken piece of rubber. It also helps if you check your belts regularly, so you can spot any problems early on – hopefully before they cause a belt to fail in the middle of a job.

Top Left
This belt has become inflexible with time, and keeps the shape of the pulleys even when it's been removed from the machine. It also has some chemical damage from exposure to coolant and solvents, and needs to be replaced with a chemical-resistant type

Top
Sectional belts are not a new idea, but modern materials are giving them a new lease of life. As their name implies, sectional belts are made up of interlinking sections. Traditionally relying on metal linkages, newer types use composite materials and can help reduce the transfer of vibration and noise from a motor. These types of belt are more expensive than ordinary V-belts, but they are useful to keep in stock because they can be quickly made to fit any length using a pair of pliers

TUTORIAL

Model and 3D-print a vase

From idea to physical vase in a matter of hours with zero 3D modelling experience necessary!





Glenn Horan

@BatGlenn13

Glenn is a software developer by day and a 3D printing enthusiast by night. Rumour has it that building and troubleshooting cheap Prusa clones is what caused him to go bald before his years.

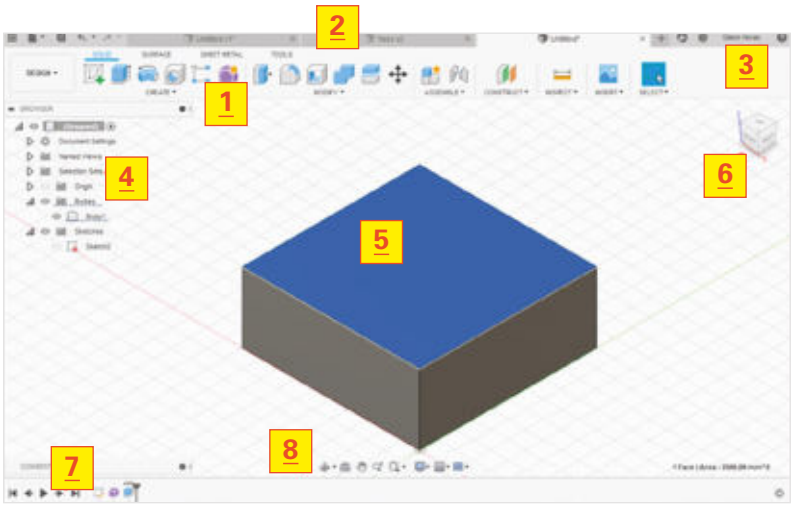
Have you always wanted to get stuck into learning how to use 3D modelling software, but resigned yourself to what's been shared already, thinking that designing your own is too difficult or time-consuming to learn? Maybe you just needed a starter project to get stuck in? If nothing else, this article aims to show you that the basics of 3D modelling software are very easy to pick up by having you follow our guide to designing and printing your first vase. Also, if you've never printed a vase before, you're in for a treat! Using Cura's spiralize outer contour mode makes printing large objects much quicker than what you'd expect, with the added bonus of using very little filament. We will also give you an introduction to the basics of Fusion 360 (polled as the most popular 3D modelling software for hobbyists in our 3D printing issue) and take you through vase designs of increasing complexity to arm you with the tools to make and hold your own unique creations, because isn't that what 3D printing is all about?

Before we get started, we need to install Autodesk Fusion 360. Simply visit here: **hsmag.cc/Xdaugd** and click 'Get Started Now' to sign up for an account, and then follow the download instructions. It's important to note that Fusion 360 is free for hobbyists (and startups earning less than $100K). Follow the installation instructions and, when you have everything set up, if it says '30-day free trial period' at the top of the screen, don't worry – you will be able to confirm that you're a hobbyist once the trial period has ended, and continue to use the software free of charge.

Now that you've signed up to and downloaded Fusion 360, let's take a quick look at the interface. Don't be too intimidated by the number of options and strange-looking icons populating the screen – you'll only need a few of them for this tutorial.

YOU'LL NEED

- **3D printer**
- **Computer able to run Fusion 360**
- **Fusion 360** (free for hobbyists)



1 **Toolbar** This is where you'll find tools such as extrude, fillet, modify, revolve etc. (We'll cover a few of these).

2 **Project tabs** You can quickly switch between projects you're currently working on.

3 **Username** Click here to access all of your preferences/settings.

4 **Browser** Here you can see and select sketches, parts, and bodies from your design.

5 **Your model** Pretty self-explanatory – the dark blue colour here indicates which face is currently selected.

6 **ViewCube** You can drag this cube or click on different faces to change the viewing angle.

7 **Timeline pane** The timeline shows all of the different sketches, shapes, and objects that make up our design. We can scan backwards and forwards through individual steps in our design.

8 **Navigation bar** Here you'll find zoom, pan, rotate, and other navigation options, but once you've learned the mouse and keyboard shortcuts, you probably won't use this too much.

Now that's out of the way, let's get stuck in! Fusion 360 and other CAD tools work by firstly making a 2D drawing, or sketch, onto a plane. This sketch is then 'extruded' into a 3D object. To draw our sketch, simply click on the first icon in the toolbar along the top of the screen (immediately to the right of the design button). You'll notice that the viewport will change slightly and a small message beside your mouse pointer is prompting you to select a plane. It doesn't really matter which plane we choose at this stage since you'll be able to rotate your model later in the slicer, but we like choosing the bottom plane (use the ViewCube for reference) as it feels natural to print a vase from the ground up. Once we've selected the face, click on the circle button on the toolbar, then click on the origin (the small circle in the middle of the screen) and move the mouse back and forth to make the circle bigger or smaller. You can make a circle of whatever size you want by clicking a second time when it's the correct size, or we can enter the diameter of the circle manually in the text box that pops up – 150mm sounds about right for our vase as this will fit on the build plate of most printers, but feel free to make this whatever size you like as we can resize it using Cura later. Finally, click on Finish Sketch in the sketch palette menu to the right (**Figure 1**, overleaf).

> "Fusion 360 and other CAD tools work by firstly **making a 2D drawing, or sketch, onto a plane**"

AUTODESK FUSION 360

Fusion 360 is a very powerful piece of software. It is used by both hobbyists and professionals alike and, as such, has a huge suite of features that would be impossible to cover in a relatively short article. In the interests of getting a quick first model under your belt, we may have glossed over some important features. If following this guide has given you the appetite to learn more, Autodesk has put together a series of hands-on video tutorials that are as comprehensive as they are easy to follow along with: **hsmag.cc/w8Xk5Z**. There are also some awesome tutorials on YouTube which are particularly useful if you're interested in specific types of modelling (character modelling, technical design, etc.)

IS IT PRONOUNCED VAYSE OR VARSE?

To make our sketch 3D, we can extrude a solid shape, like a cube, or we can extrude what's called a 'form'. With forms, we've access to all of the different edges, vertices, and faces that the 3D shape we're making will have. To create a form, click on the inside of our newly drawn circle so that it turns blue, then activate the 'Solid' tab from the toolbar. From here, click on the 'create form' →

QUICK TIP

You can find system requirements for Fusion 360 here: **hsmag.cc/rrmB2E**.

TUTORIAL

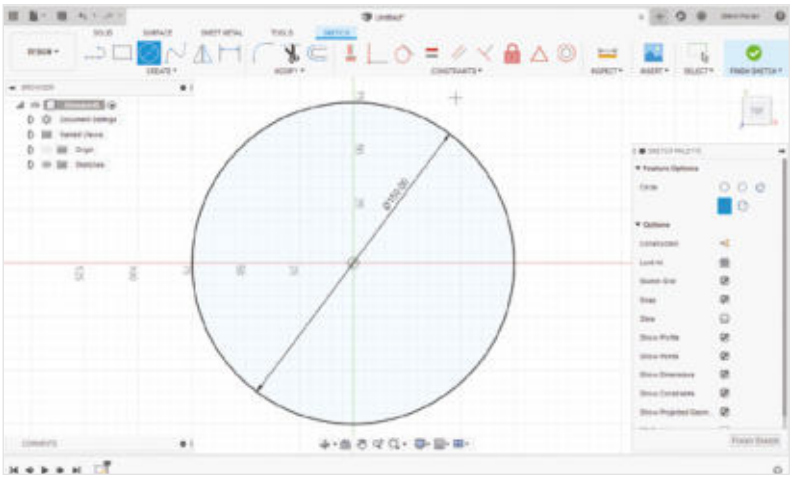


Figure 1
Once we've drawn our sketch, we get access to the sketch palette menu on the right to enter more parameters

button (the purple cube with rounded edges) and note that our viewport has shifted perspective so as to better visualise 3D space. The toolbar now shows us a variety of different shapes, but we want to extrude from our circle: click on 'Create' then select Extrude from the drop-down menu that's just been revealed (**Figure 2**). We can now click on the arrow that's appeared in the middle of our circle and drag it up to create a cylinder from our circle. Similar to drawing our sketch, we can define the exact height of our first cylinder. Let's make it around 30 mm, and then click 'OK' in the Extrude menu to the right.

THAT'S A FUNNY-LOOKING VASE

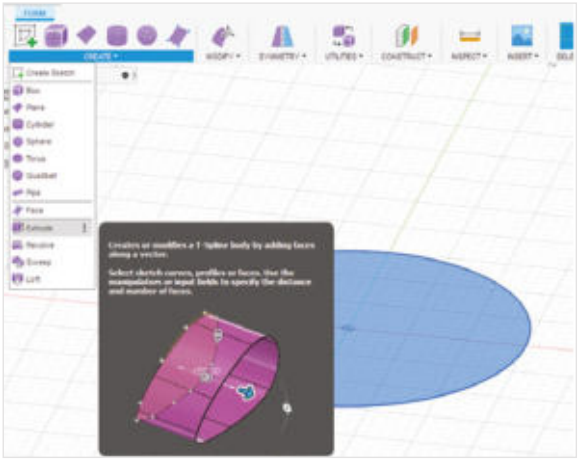
You may be wondering why we started with such a short cylinder, but if we double-click on the top rim of our cylinder (it will change colour) and then click the icon above Modify in the toolbar above, you'll see that a series of icons and tools have appeared in the centre of our cylinder (we'll call these the form tools – see **Figure 3**) and the mouse pointer is prompting us to hit 'Alt key to extrude'. That's exactly what we're going to do – hold the **ALT** key and drag the upwards-pointing arrow to extrude, by another 30 mm. The key difference between doing this and just starting with a bigger cylinder is that now there is a line running horizontally around our cylinder that we can select and modify. Repeat this process until you have six vertical segments in the cylinder. You don't need to click Modify and select the top edge of our cylinder each time: just hold **ALT** and drag the upwards-pointing arrow for each new segment.

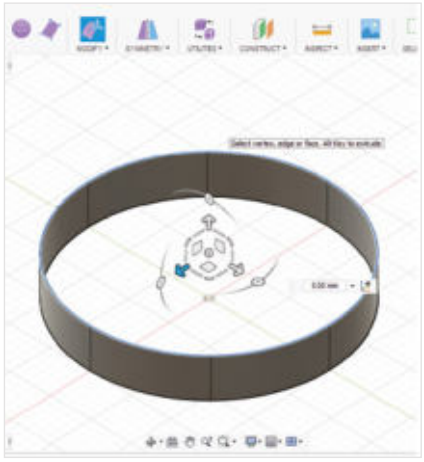
Now let's turn our cylinder into a vase. This is the part of the tutorial where you can get creative, but we'll give you our parameters in case you want to create our exact design. We can select any of the horizontal rings through the cylinder that we created, so let's start by double-clicking on the very bottom one (the one touching the plane). If it isn't still selected, click Modify in the toolbar to get access to all of our form tools again. This is the fun part – click on the small hollow circle in the very centre of the form tools and drag your mouse left and right. The bottom of your cylinder should start growing and shrinking with your mouse movements (see where we're going?). You can also enter a value here into the text box that pops up if you click on the hollow circle instead of click-dragging. To stick with our design, enter 0.75 here, which will shrink the bottom segment of the cylinder to 75% of its original size. Moving up the vase, leave the next two segments as they are, set the third to 0.5, the fourth to 0.25, the fifth to 0.5, and finally the top edge to 0.75. It should look like **Figure 5**. If you make any mistakes, you can use the Undo button in the top-left of the screen. While you should absolutely go nuts here making your own design, be very careful at this stage not to get too carried away

> "The bottom of your cylinder should **start growing and shrinking with your mouse movements**"

Figure 2
Forms are hollow, rather than solid, shapes

Figure 3
It's sometimes useful to rotate your view so that you can see the form tools better. Also notice that the entire top edge of the model is selected





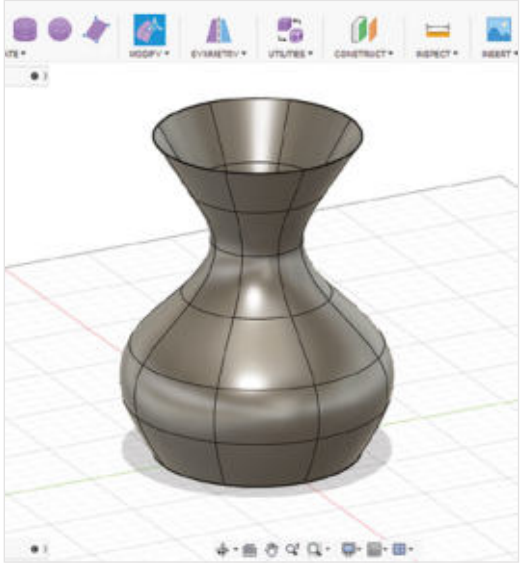
FILAMENTS FOR VASES

Choice of filament for your vase can make a huge impact. While vases look great regardless of what filament is used, there are some which can give some really cool effects. We love marble effect filament to make your thin plastic vase look like solid stone (from a distance). Wood effect can make it look like your vase is actually a piece of brown paper origami. 'Silk' filaments give vases a really awesome shine that can look downright trippy with the right vase design. Check out this example from 'riskable' on Reddit, who used rainbow colour silk filament to make his vase look almost pixelated: **hsmag.cc/QUeNnH**.

and make steep angles and large overhangs for your print – we're going to be using a special vase mode in Cura to print our creations very quickly, and if we need supports, it'll ruin the effect. When you're happy with your design, click on Finish Form – the big green tick in the top-right of the screen.

Believe it or not, we're finished with the modelling of our basic vase. Even though there is no floor to our model, Cura will add this with the vase mode we'll be using later. Next, we need to export our model as an STL file so that our slicer will recognise it. Fusion 360 requires that we save our file first – click the Save icon and don't worry for now about the different prompts that pop up, as these are for collaboration features. Next, click on

Figure 5
Not too shabby for a first vase



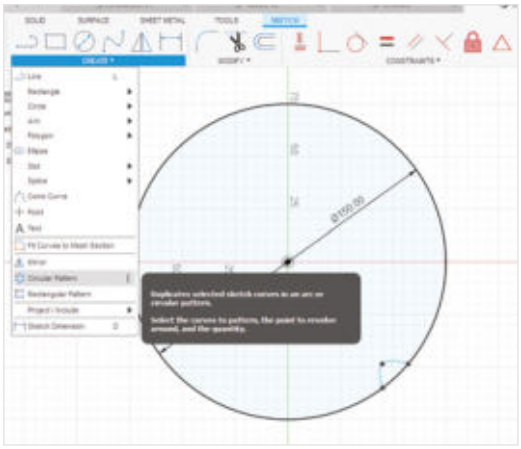


Figure 6
Experiment with cutting different shapes into your circle. Some can look downright Lovecraftian after they've been extruded

File, then Export at the top-left of the screen – in the 'type' drop-down menu, scroll to the bottom and select .stl. You can select a location that it will be saved on your computer, then click on Export. The export step may take a few minutes – you can use this time to boot up Cura and start preheating your printer!

CURA

Finally, we need to slice our model and get it printing. We're going to use Cura (the latest version at the time of writing is Cura 4.4) as our slicer, as it's incredibly popular and has an option for spiralize outer contour mode. First, import the model as you normally would import any STL file, and take a second to slice the model and look at how long it'll take to print (for us it was ~36 hours). Next, go into your print settings tab, click on Custom, search for 'spiralize outer contour', and click the checkbox next to the setting. Re-slice your model and check out the time difference – ours will now print in 4 hours, 39 minutes! There are a couple of considerations for using spiralize outer contour – it's even more important than with normal printing models that you have the temperature dialled in correctly for →

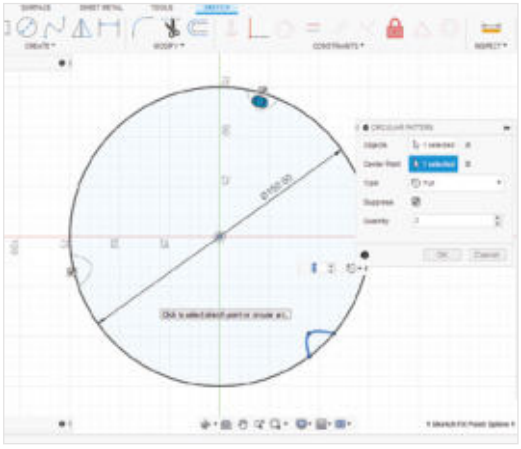


Figure 7
Once you've selected the shape and the origin, the circular pattern menu will say '1 selected' for each of these options, and you'll be able to make your repeated pattern

QUICK TIP

SHIFT-clicking in your mouse scroll wheel will allow you to rotate around your model.

SPIRALIZE **OUTER CONTOUR**

In case you missed our Cura guide, this print parameter will print a hollow version of your model with a floor, walls one layer thick, and no roof – it's easy to see why it's so popular for printing vases. While you may see it referred to as vase mode, it's called 'spiralize outer contour' because your printer will print the walls as one continuous line by very slowly increasing the height on the Z-axis (picture a very tight corkscrew), rather than generating height by printing layer by layer as normal.

QUICK TIP

You can click on the Design button on the top-left of the UI to access 'Render' mode to make your vase look awesome on screen.

the filament you're using. You may want to print slightly slower than usual to give each layer time to cool properly. While this is true of any print, it's even more important when the walls are only one layer thick. We're using 50 mm/s on a Creality Ender 3, but feel free to experiment with what works for you.

Now, let's make it more interesting. Save your vase and click on the plus icon in the tabs section of the screen to start a new project. Follow the same steps as previously until we have a 150 mm diameter circle drawn. We're now going to add a repeating pattern to our sketch so that when it's extruded, we have a texture on the surface of the vase. Click on the 'fit point spline' icon (the wavy line) in the toolbar under the sketch heading. Next, click anywhere on the outer perimeter of your circle and move the line you're drawing a few millimetres into the centre of the circle. Left-click and then click again on the perimeter of the circle about 1 cm away from the original point, and finish by clicking on the green tick. You should now have a small semicircular indentation in our sketch, like that shown in **Figure 6**.

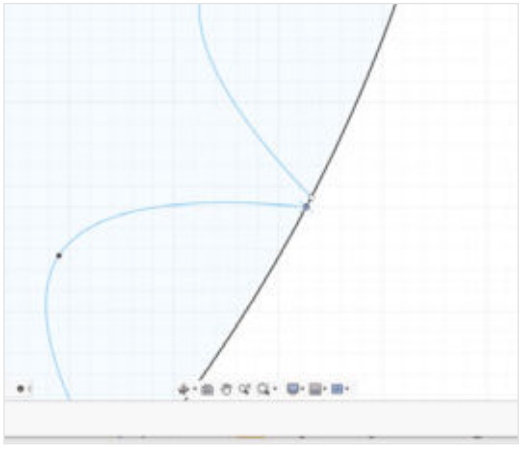
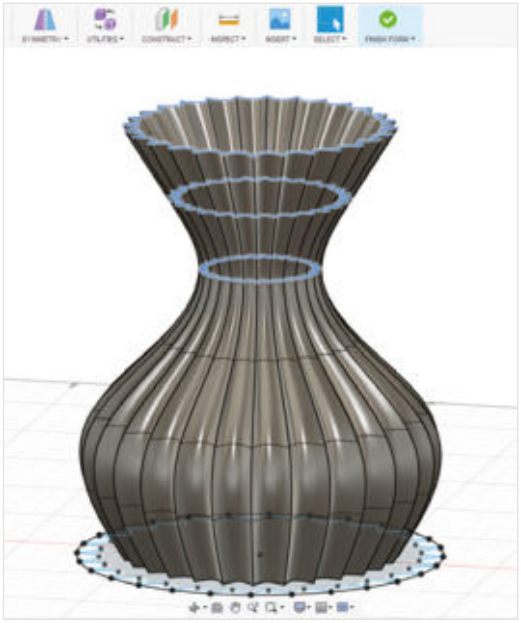


Figure 8
As you drag the two small circles closer, they should snap together

Figure 9
The top three lines in our vase are selected, and we can access our form tools again by clicking Modify from the toolbar



SPLINE-TINGLING DESIGN

To multiply this shape the whole way around our vase, we need to click on the Create button in the sketch toolbar, scroll to the bottom, and select 'Circular Pattern'. You should see the circular pattern menu appear beside our circle. Select 'Objects' (the pattern to be repeated) from the circular pattern menu, click on the line you've just drawn, then select 'Center Point' from the same menu and click the origin (the small circle in the centre of our sketch). You should see that there are now three of

> "You should see that there are now three of the **original shape we drew in the wall of our circle**"

the original shape we drew in the wall of our circle, and another option has appeared in our circular pattern menu – quantity (see **Figure 7**). Keep increasing the quantity until the shapes are almost touching (but not overlapping), and then click OK. For our design, this was 28, but it may be slightly different for yours. Now zoom right into the area

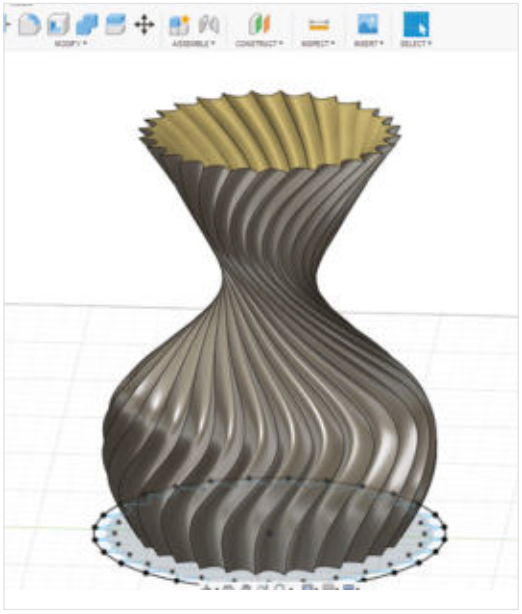


Figure 10
Our design has a nice wide base for the vase to sit flat, but it may look more 'vase-like' if you make the bottom segment smaller

where two of the lines almost touch and join the two shapes by dragging one of the small circles at the tip of a semicircle onto the small circle of the next semicircle along (see **Figure 8**). Finally, zoom out again and follow the steps for the previous vase to extrude a form and then create our vase shape again.

A LITTLE TWIST

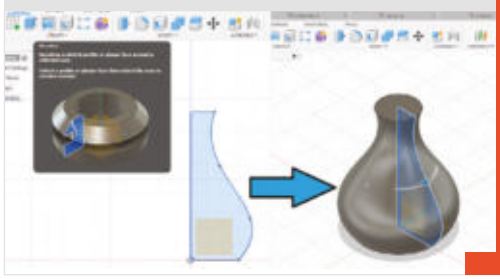
The last thing we'll do with this vase is to add a nice twist to the pattern on the surface. We can do this using the same form tools we use to increase the thickness of different sections. We can also select multiple segments at once for this part: before clicking the green Finish Form tick after changing the ring widths, double-click on the top edge of your vase (if it isn't already selected). Then, while holding the **SHIFT** key, click on each of the next two lines down so that your vase looks like **Figure 9**. Now, click Modify on the toolbar to get our form tools available again. In our form tools, you should see three curved lines with circles in them. These allow us to rotate our form around different axes. Find the one that's in the same plane as the top of our vase, then click and drag it left or right to twist the vase. We've used 85 degrees, but feel free to twist different sections until you're happy with the look of it. It's important not to over-twist things here – if any of your lines overlap, then things won't work further down the process. Finally, click on Finish Form again and you have your finished vase – check out ours in **Figure 10**!



Above
A collection of vases designed and printed by the author, complete with flowers!

A CHEAT'S VASE?

There are many ways to make a vase in Fusion 360. The quickest is to make a quick sketch of a cross-section of half a vase (see the picture below). Then, in the toolbar, click on the rotate button. The rotate menu will pop up and will require that we select an axis, just click on the line on the inside of our vase sketch, then click OK. This will rotate your sketch into a full 3D model. Note that even though this is a solid object, if we select spiralize outer contour in Cura, it will print it as a hollow vase.



QUICK TIP

Play with cutting different shapes into your starting shape and extruding them to see how they look.

BOOK OF MAKING

VOLUME 2



ONLY £10

WHSmith

BARNES&NOBLE

Available on the App Store

Google Play

THE BEST PROJECTS FROM

HACKSPACE MAGAZINE

THE ULTIMATE SKILLS, TRICKS, AND MAKES

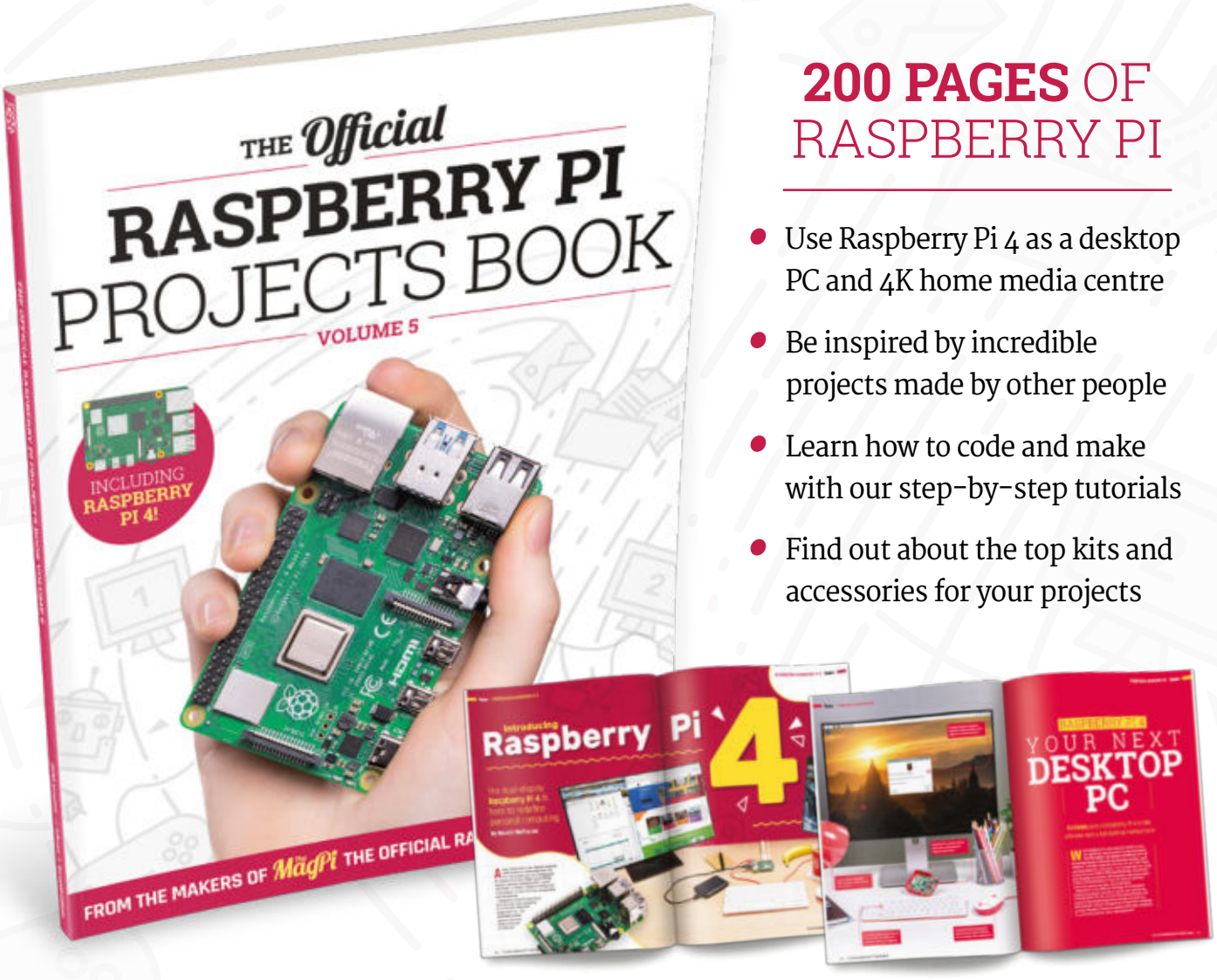
AVAILABLE NOW

hsmag.cc/store

FROM THE MAKERS OF **HackSpace** MAGAZINE

THE *Official* RASPBERRY PI PROJECTS BOOK

VOLUME 5



200 PAGES OF RASPBERRY PI

- Use Raspberry Pi 4 as a desktop PC and 4K home media centre
- Be inspired by incredible projects made by other people
- Learn how to code and make with our step-by-step tutorials
- Find out about the top kits and accessories for your projects

Buy online: magpi.cc/store

TUTORIAL

Real-time Twitch gaming

Low-latency interfacing requires a kernel driver, right? Think again



Andrew Henderson

Dr. Andrew Henderson is an author, researcher, and software engineer. He has developed Linux embedded and desktop software for over 20 years. His website is **icculus.org/~hendersa**.



Figure 1 SNES gamepads have eight buttons and a directional-pad, making them a flexible input option for all sorts of projects (Credit: adafruit.com)

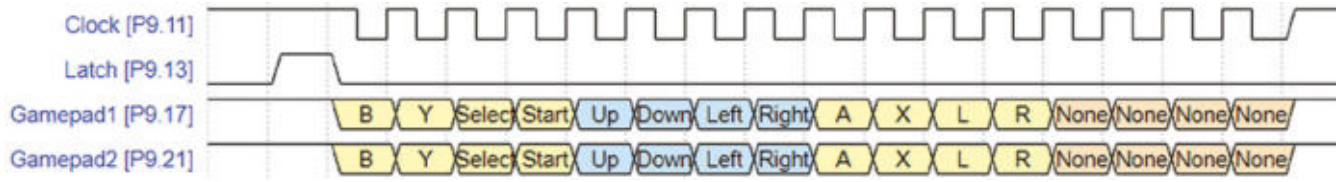
Linux is available for most ARM single-board computers, and the wide variety of open-source software and tools makes it easy to hack projects together. You don't need to worry about the low-level details of the system because the Linux kernel hides those away. You can concentrate on what you want to do, without worrying about every single detail of how you're going to do it.

It would be nice to get the performance and hardware access of a kernel driver without having to actually write one. You could treat your platform like a Linux system to do all of the hard stuff (video, audio, networking, memory management, etc.), and treat it like a microcontroller to do the bare-metal, time-sensitive GPIO interfacing. Is such a thing possible?

In this article, you'll see how to use the programmable real-time units (PRUs) of the BeagleBone Black (BBB) to read from and write to GPIOs directly. You'll use a PRU to natively interface the GPIOs to a pair of Super Nintendo Entertainment System (SNES) gamepads (**Figure 1**) for lightning-fast, low-latency input… all without using a kernel driver!

INS AND OUTS OF PRU INTERFACING

What exactly is a PRU? A PRU is an independent RISC coprocessor with its own memory and



registers. The TI AM3359 processor of the BBB contains a pair of PRUs. The PRU instruction set has arithmetic and logical operations (add, subtract, shifts), loads/stores for registers and memory, and control flow operations (compares, loops, jumps). Execution is very fast, taking only one 5 ns clock cycle to execute an instruction. PRU firmware size is limited to 8kB, but a program that toggles GPIOs will easily fit within only a few dozen bytes.

The BBB has 33 PRU-enhanced GPIOs available: 16 for output and 17 for input. These GPIOs can be accessed by a PRU in only a single PRU clock cycle! However, multiplexing enhanced GPIOs to the BBB's P8 and P9 headers often requires unmuxing other BBB features that you might want to use (like audio, video, and SPI channels). For applications where enhanced GPIOs create conflicts, the PRU can still interact with standard GPIOs. This is a bit slower, though, taking three to four PRU cycles per each read/write of a GPIO control register. For this project, you'll be using standard GPIOs for flexibility.

The Linux kernel uses two methods for communicating with PRUs: remote procedure calls (remoteproc, which is unrelated to POSIX RPC) and userspace I/O (UIO). Remoteproc provides communication between a custom Linux kernel driver and the PRU firmware. UIO skips the kernel driver and allows user space applications direct access to memory shared with the PRU.

Which PRU interface method should you use? This article uses UIO, which is very straightforward and simple. Remoteproc is far more powerful and efficient than UIO, but it requires that you write a kernel driver to use it. UIO is a good place to start for beginners. If you're doing complex projects in the future, consider using remoteproc.



BITS AND BYTES OF THE GAMEPAD PROTOCOL

SNES gamepad connectors have seven pins, though only five are used to interface with the gamepad. Two pins provide power (5 V and ground), and two input pins accept 'clock' and 'latch' control signals. The final pin is an output 'data' signal which provides a serial stream of data (1 bit per pulse of the clock signal) that represents the state of each gamepad button and the directional-pad (D-pad).

> "Useful data is only returned during **the first twelve of the clock signal pulses**"

Figure 2 shows the timing diagram for the serial communication protocol with two gamepads. Each time that you wish to sample the gamepad state, the latch signal is raised for 12 µs and then lowered to wake up the gamepad. After the latch pulse, a series of 16 pulses, each 12 µs in duration, are sent on the clock signal. During each clock pulse, the state of one gamepad button is returned on the data line as a bit of data. The bit for that button will be 0 (low) if the button is pressed and 1 (high) otherwise. There are only twelve bits (eight for buttons and four for the D-pad), so useful data is only returned during the first twelve of the clock signal pulses. During the final four clock pulses, a 1 is returned on the data signal. After these 16 pulses, the gamepad goes back to sleep until another latch signal is received.

Interfacing with gamepads using this protocol is an ideal PRU starter project because the protocol is →

Figure 2
For more details on the SNES gamepad protocol, visit hsmag.cc/JAQwz7

Below
Relive your youth with our nostalgic games controller

YOU'LL NEED

- **BeagleBone Black**
- **One or two Super Nintendo gamepads** (adafruit.com, Part #131)
- **One or two female Super Nintendo gamepad connectors** (raphnet-tech.com, Part #250025)
- **Four bidirectional BSS138 line-level converters** (adafruit.com, Part #757)
- **Breadboard and jumper wires**
- **Basic soldering equipment**

RECOMMENDED

- **4GB or larger microSD card**

QUICK TIP

For step-by-step instructions on updating your BBB, downloading and writing images to microSD cards, logging into your BBB, and configuring your BBB's network connection, visit **beagleboard.org/getting-started**

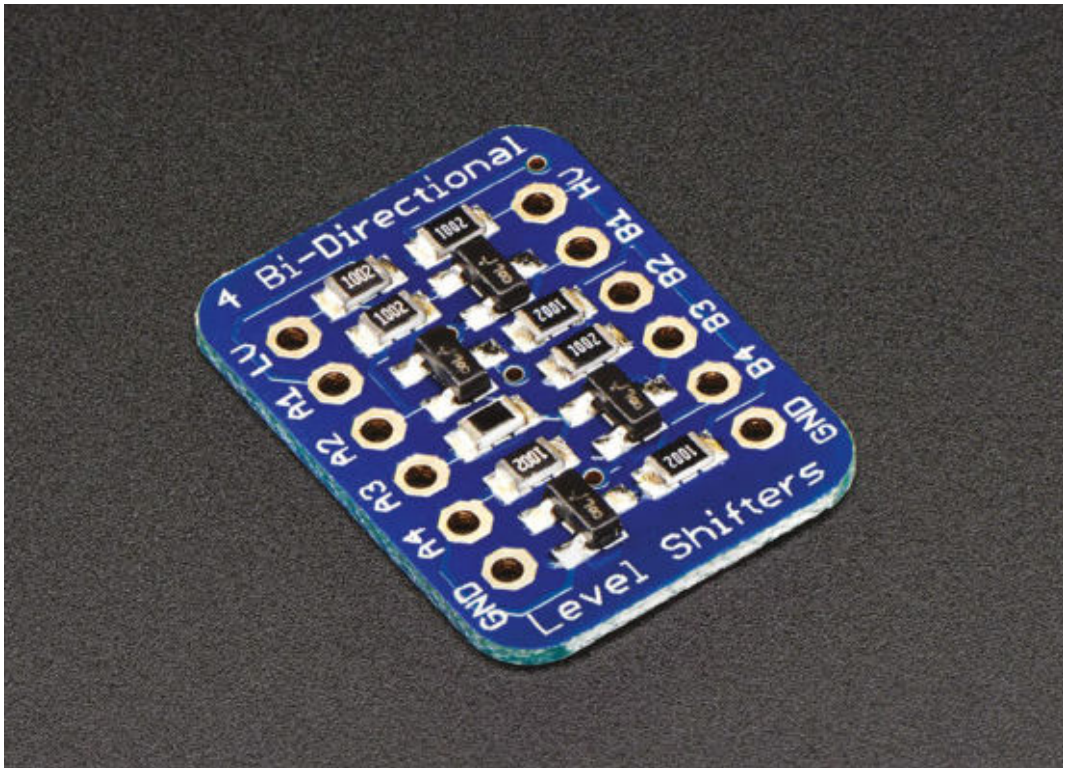


Figure 3 Double-check your breadboarding to ensure that your BBB's GPIO pins are only connected to the low-level side ('A'-side) of your logic-level converter PCB (Credit: adafruit.com)

simple, implemented solely using GPIOs, and has strict timing constraints (but not so strict that you are limited to using PRU-enhanced GPIOs).

PREPARING FOR PRUS

Working with PRUs requires a BBB system with PRU support in its kernel, and a toolchain for building PRU firmware. You should start experimenting with PRUs using a known-good BBB OS image with PRU support that runs from a microSD card. The recent Debian 9.9 Internet of Things (IoT) image available for the BBB provides a 4.14 Linux kernel and root file system that supports both the remoteproc and UIO methods of PRU interfacing. If you already have a BBB OS environment that you like, feel free to try it when following these instructions. However, troubleshooting the setup of the PRU can be difficult, so we recommend using the suggested IoT image to make things easier.

By default, the IoT image is configured to use the remoteproc PRU interface. Since you'll be using the UIO interface instead, you'll need to make a small change to the image. The U-Boot bootloader configuration script (**/boot/uEnv.txt**) on your microSD must be changed to enable UIO. The image comes with the Nano, Pico, and Vim text editors already installed, so you should be able to find one that you like to edit **uEnv.txt** on your running BBB system.

Log into your BBB system and change the `uboot_overlay_pru` setting of **uEnv.txt** to use the UIO device tree overlay, rather than the remoteproc one. To make the edit, simply insert a hash (`#`) at the start of line 38 to comment out the `pru_rproc` overlay option for (`4.14.x-ti kernel`) and remove the starting hash on line 42 to comment in the `pru_uio` overlay option. After the change, the file should look like this (starting at line 37):

```
###pru_rproc (4.14.x-ti kernel)
#uboot_overlay_pru=/lib/firmware/AM335X-PRU-RPROC-4-14-TI-00A0.dtbo
###pru_rproc (4.19.x-ti kernel)
#uboot_overlay_pru=/lib/firmware/AM335X-PRU-RPROC-4-19-TI-00A0.dtbo
###pru_uio (4.4.x-ti, 4.14.x-ti, 4.19.x-ti & mainline/bone kernel)
uboot_overlay_pru=/lib/firmware/AM335X-PRU-UIO-00A0.dtbo
```

Once **uEnv.txt** has been modified, save your changes, sync the file system, and reboot your BBB:

```
$ sync
$ sudo shutdown -r now
```

With the system rebooted, log into your BBB once more and ensure that the UIO PRU kernel module is properly loaded on your system. The easiest way to do this is to check for the presence of the **/dev/uio0** file, which is created when the UIO driver is loaded:

```
$ ls -l /dev/uio0
crw-rw---- 1 root users 246, 0 Sep  3 02:28 /dev/
uio0
```

If you do not see the **/dev/uio0** file, double-check your edit to the **uEnv.txt** file to ensure that the UIO overlay is commented in and that all RPROC overlays are commented out. With the UIO PRU driver loaded, you are now ready to begin experimenting with the PRU via the UIO interface.

The PRU development package from **beagleboard.org** contains the PRU assembler (PASM), which compiles PRU assembly code into binary PRU firmware images. Download and build the source code for PASM from GitHub onto your BBB:

```
$ cd ~
$ git clone https://github.com/beagleboard/
am335x_pru_package
$ cd am335x_pru_package/pru_sw/utils/pasm_source
$ ./linuxbuild
$ sudo cp ../pasm /usr/local/bin
```

Now that you have a working system containing the tools needed to begin working with the PRU, you can start interfacing the BBB with some hardware.

CREATING THE GAMEPAD INTERFACE CIRCUIT

Unfortunately, SNES gamepads use 5 V logic-level control signals, and BBB GPIOs use 3.3 V. You need line-level converters to translate between the two. Because this project has two output signals (latch and clock) and two input signals (the data line from each gamepad), it is simplest to use bidirectional converters when constructing your interface circuit. Adafruit sells a PCB with four BSS138 bidirectional converters mounted on it (**Figure 3**) which is perfect for this project. Adafruit's PCB has two sides: one is low voltage (the 'A'-side), and one is high voltage (the 'B'-side). Any low-voltage signal connected to pin A1 will be translated to the high-voltage signal connected to pin B1 and vice versa. A2 maps to B2, A3 to B3, and A4 to B4.

The SNES female connectors have seven thick pins that are designed to anchor the connector in place when soldered to a PCB. These pins are much wider than the diameter of the holes in a breadboard. Solder a breadboard wire to each pin (skipping the unused pins 5 and 6) to make it easy to wire each connector into the breadboard (**Figure 4**).

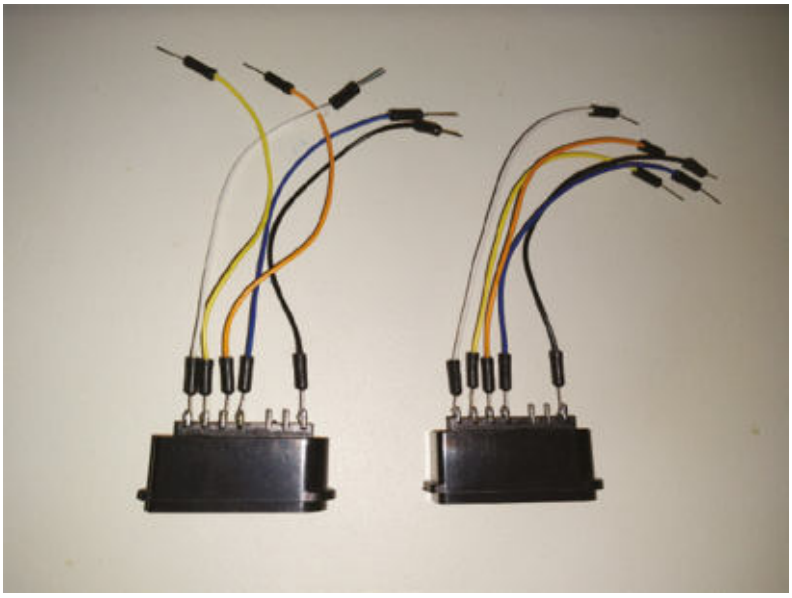


Figure 4 Adding the breadboarding wires makes it easy to reuse the connectors in multiple projects

This firmware uses four GPIO pins that are muxed to the BBB's P9 connector: latch is assigned to pin P9.11, clock to pin P9.13, gamepad one data to pin P9.17, and gamepad two data to pin P9.21. These four GPIOs were selected because they are muxed as GPIOs by default in BBB OS images. **Figure 5** (overleaf) shows the completed interfacing circuit.

> "It is simplest to use bidirectional converters **when constructing your interface circuit**"

Wiring the circuit is easy when using a breadboard. Both the gamepads and the line-level →

THE SIGNALS ON THE PINS OF A SNES CONNECTOR

Pin 1: +5 V	Pin 2: Latch
Pin 3: Clock	Pin 4: Data
Pin 5: Unused	Pin 6: Unused
Pin 7: GND	

QUICK TIP

For more information on the PRU instruction set, and many other details of PRU interfacing, visit **hsmag.cc/eqGJXf**.

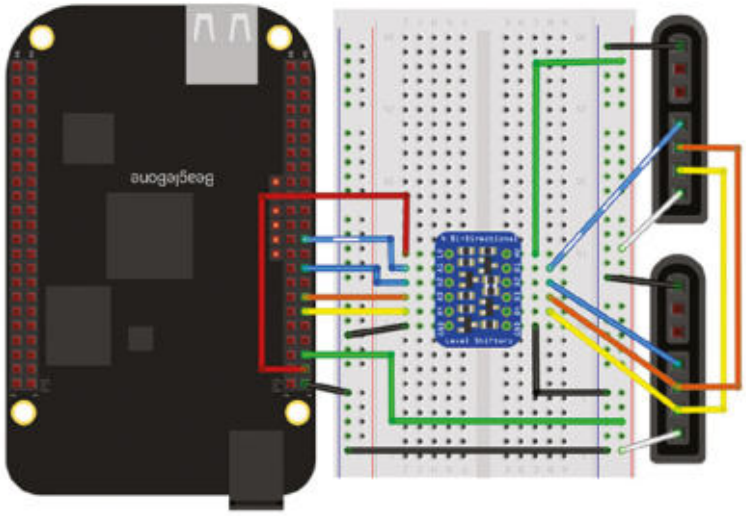


Figure 5
The interfacing circuit, complete with line-level conversion between the gamepads and the BBB

converter PCB use the 5V and GND signals from the BBB, so wire your BBB's P9.5 (5V) to the breadboard VCC bus and P9.1 (GND) to the breadboard ground bus. This way, both the PCB and the controllers have easy access to 5V and GND.

Wire the BBB to the line-level converter PCB by connecting P9.3 (3.3V) to LV, P9.11 (latch) to A1, P9.13 (clock) to A2, P9.17 (gamepad one data) to A3, P9.21 (gamepad two data) to A4, and your breadboard's ground bus to GND. On the high voltage side of your PCB, connect HV to the 5V VCC bus of your breadboard and GND to the ground bus. Wire the PCB to the gamepad connectors by connecting B1 (latch) to pin 2 of both connectors, B2 (clock) to pin 3 of both connectors, B3 (gamepad one data) to pin 4 of one gamepad connector, and B4 (gamepad two data) to pin 4 of the second gamepad connector.

FIDDLING WITH FIRMWARE

Now that you have hardware connected to the BBB, it is time to get some PRU firmware talking to it. Fetch the prugamepad project from GitHub onto your BBB. It contains the example code used in this article:

```
$ cd ~
$ git clone https://github.com/hendersa/
prugamepad
$ cd prugamepad
$ make
```

The PRU code implementing the gamepad protocol is in the source file **gamepad.p**. It uses only 61 assembly instructions, and the assembled firmware binary is only 244 bytes in size! There isn't much variety in the instructions being used, which shows that you don't have to understand every single PRU instruction to make useful firmware programs.

gamepad.p is divided into four main pieces: defining useful constants (lines 14–27), initialising shared memory (lines 30–47), polling gamepad button states (lines 53–110), and busy-wait delays for timing (lines 112–126). Don't worry about the details of shared memory initialisation, as this is boilerplate code provided by TI. It configures the PRU to share a memory location with Linux user space.

After PRU initialisation, the firmware's main loop begins and runs forever. This snippet of the main loop shows the basic mechanics of how the PRU handles setting GPIO outputs:

```
// Set latch
LBBO r0, GPIO_OUT, #0, #4
SET  r0.LATCH_BIT
SBBO r0, GPIO_OUT, #0, #4
```

First, the value held in a memory-mapped 32-bit GPIO output control register (`GPIO_OUT`) is loaded into a PRU register via the `LBBO` instruction. Each bit in this control register represents the output state of one GPIO pin, and these bits are either set to 1 using the `SET` instruction or cleared to 0 using the `CLR` instruction. Once the bits have been manipulated, the changed value is stored back into the control register via the `SBBO` instruction. When new state data is written into the control register, the physical states of each GPIO change immediately.

It is a similar process for input GPIOs. Each bit in a 32-bit GPIO input control register (`GPIO_IN`) contains bits representing the current state of input GPIOs. If a GPIO pin is high (connected to 3.3V), that GPIO's bit in the control register is set to 1. If that GPIO is low (connected to GND), the corresponding bit in the control register is cleared to 0. This snippet of the main loop shows the reading of GPIO input values.

```
// Read bit from each gamepad
LBBO r0, GPIO_IN, #0, #4
LSR  GP0_REG, r0, GP0_BIT
```

QUICK TIP

The **am335x_pru_** package contains the source code to prussdrv. The prugamepad Git repo also includes a copy.

CODE MADE CLEARER

There aren't PRU registers with names like 'GPIO_IN' and 'GPIO_OUT'. PASM provides a '#define' preprocessor directive for you to give convenient, easy-to-understand names to registers and constants. Use this to make your assembly code easier to read and understand.

```
AND  GP0_REG, GP0_REG, #1
LSR  GP1_REG, r0, GP1_BIT
AND  GP1_REG, GP1_REG, #1
```

The **LBB0** instruction reads all four bytes of **GPIO_IN** and copies it into a PRU register. Logically shifting right and masking (using the **LSR** and **AND** instructions) isolates a specific GPIO's bit. Once the bit is isolated, it is tested to see whether it is 1 or 0. By repeating this process of sampling and shifting, the state of all gamepad buttons and the D-pad are collected and then written out to a shared memory location.

TALKING TO THE PRU FROM USER SPACE

Now that you have firmware that talks to hardware, you'll need user space code that talks to the firmware. Such user space code has two important jobs. First, it must manage the PRU program life cycle: load firmware into a PRU, begin PRU execution, and stop PRU execution (if needed). Second, it must access memory shared with a PRU to collect any data that the PRU generates.

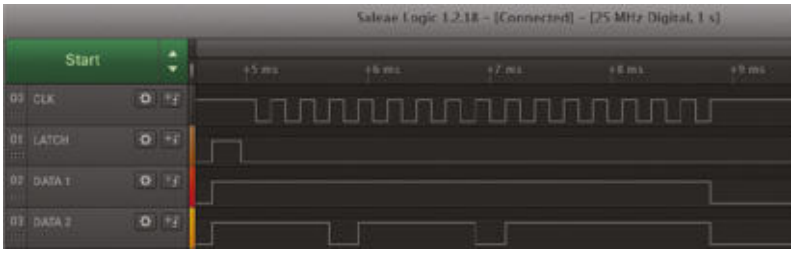
A very simple PRU interfacing program looks like this:

```
int main(void) {
  PRUSetup("./firmware.bin");
  do {
    printf("Shared memory: 0x%08x\n",
PRUState());
  } while (1);
PRUShutdown();
}
```

PRUSetup() loads the firmware onto a PRU and tells the PRU to begin execution. Likewise, **PRUShutdown()** stops the PRU's execution. **PRUState()** is called to repeatedly poll memory shared with the firmware. As the firmware runs, it continually updates the value held in shared memory.

It may seem like a lot of complex details are hidden inside those PRU functions, but there really aren't. TI provides a small, BSD-licensed C library called 'prussdrv' that performs PRU life cycle management. This makes implementing these PRU* functions simple since they are largely wrappers around calls to prussdrv code.

prugamepad implements a complete **PRUSetup()** and **PRUShutdown().** It has a more complex sampling loop that interprets the value of each bit retrieved from shared memory. Any change in individual bits between two successive samples indicates that a gamepad button or D-pad has changed state. Try running it to see it in action:




```
$ cd ~/prugamepad
$ sudo ./setup.sh
$ sudo ./prugamepad
```

As prugamepad runs, the current state of both gamepads is dumped to the screen every time that the value held in shared memory changes. You can use a logic analyser or an oscilloscope to watch the gamepad protocol signal activity on the BBB's pins (**Figure 6**).

WHAT'S NEXT?

Now you've seen some of the basics of what PRUs can do, it is time to start experimenting! PRU-based projects include flying drones, driving 3D printers, acting as digital logic analysers, and controlling robots! Visit **hsmag.cc/CmQxjd** to learn more about how to use PRUs in your own projects.

Figure 6
In any digital interfacing project, a logic analyser is invaluable when troubleshooting and debugging your software and firmware

Below
The SNES is the greated games console of all time [citation needed]



DON'T MISS THE **BRAND NEW** ISSUE!

SUBSCRIBE FROM JUST £5

- **FREE!** 3 issues for the price of one
- **FREE!** Delivery to your door
- **NO OBLIGATION!** Leave any time

FREE PI ZERO W STARTER KIT*

With your 12-month subscription to the print magazine

magpi.cc/12months

* While stocks last

Buy online: store.rpipress.cc

HackSpace
TECHNOLOGY IN YOUR HANDS

FIELD TEST

HACK | MAKE | BUILD | CREATE

Hacker gear poked, prodded, taken apart, and investigated

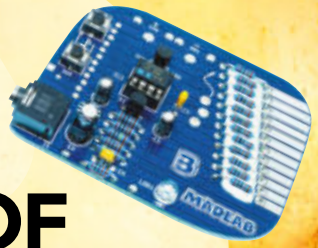
PG 112



DIRECT FROM SHENZHEN:
COMPONENT TESTER

Find out what's in your spares stash without having to use a magnifying glass

PG 114



BEST OF BREED

Electronics project kits for makers young and old

PG 120



CAN I HACK A COIN SLOT?

Make millions by adding a payment feature to your makes

REVIEWS

122 **SL1 and CW1**
Prusa's new resin system

126 **Pokit**
A portable multimeter

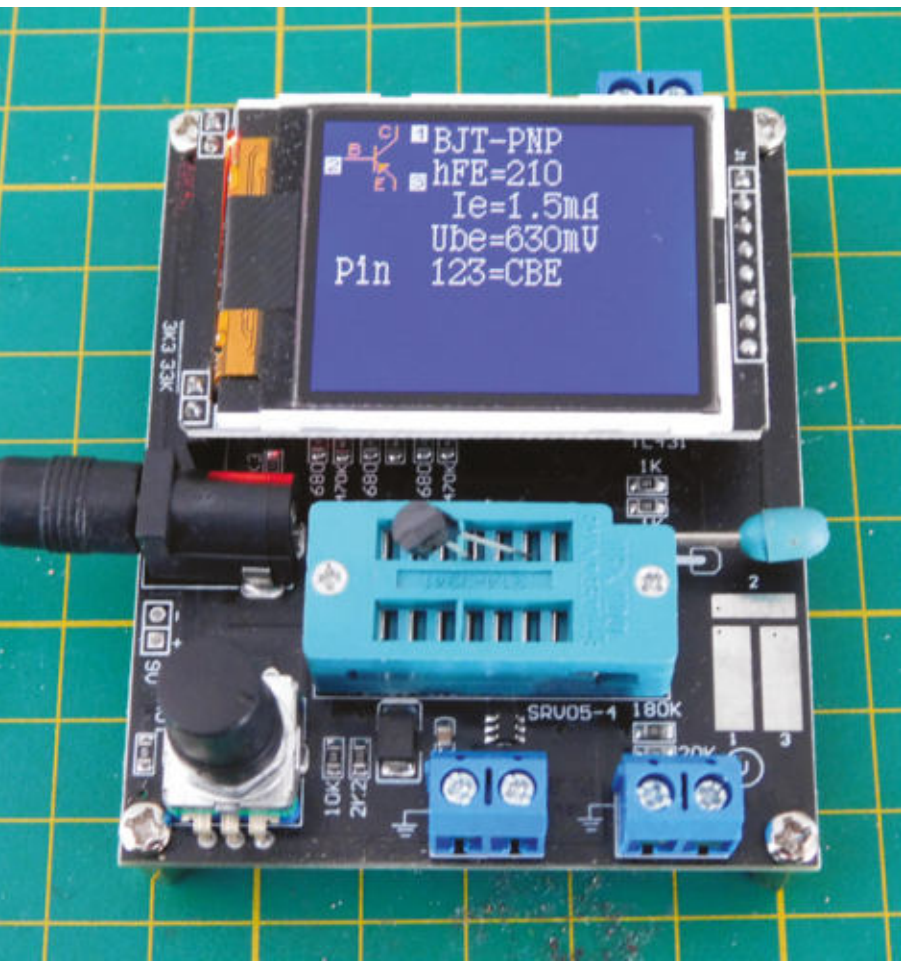
128 **AmbiMate MS4**
Many sensors, one footprint



Component testers

Discover what transistors you have jumbled together in your parts box

By Ben Everard @ben_everard

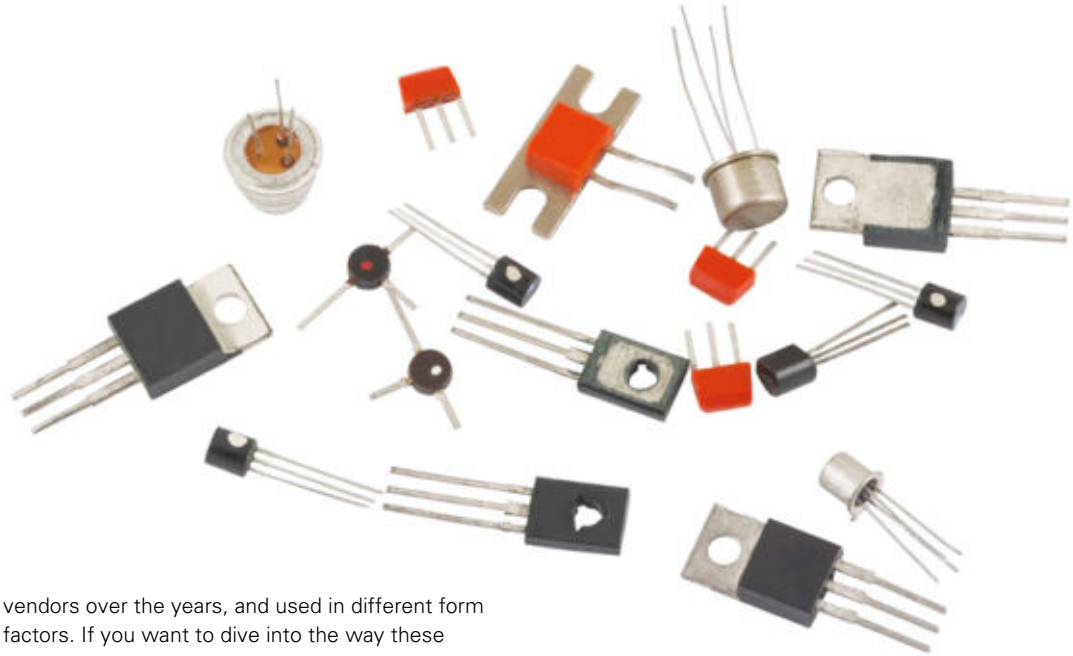


Below
Plug in your components, hit go and, voilà, you have your ID

Let's be honest, dealing with electronics means dealing with lots of little things. These little things all have bits of information we need to know about them, but very little space on which to put it. Resistors have solved this problem by using a system of coloured bands that let you know the value and tolerance of a particular item, but not all components are quite this friendly. What's that transistor you have? If you've got good eyesight, you'll be able to read the product code, but that doesn't tell you much, unless you happen to know what it is. Even more difficult is knowing whether a component still works. If it's charred, black, and smoking, then there's a pretty good chance it's now dead, but it's often not easy to tell.

Fortunately, there are component testers to help us here. We picked up a 'GM328 Mega328 Transistor Tester LCR Diode Capacitance ESR Voltage Frequency Meter PWM Square Wave Frequency Signal Generator' from SeeSensor on AliExpress for £7.93, including UK delivery. It came without a power supply, but takes 7 to 12 V from a standard 2.1×5.5 mm power plug. It's quite common to run these off 9 V batteries, and component testers typically have very low power usage, so batteries should last a very long time.

As you may guess from the name, this is built on the ATmega328P microcontroller – the same one found in the Arduino Uno. It's likely that the firmware in this device is built on the open-source code originally created by Markus Frejek, and later adapted by others. This firmware has been used by many



Left
Find out what all those spare parts you have really are

vendors over the years, and used in different form factors. If you want to dive into the way these component testers work and find out more about the processes behind them, there is a great YouTube video on the subject from Andreas Spiess at **hsmag.cc/runVun**. It's well worth a watch, as the process can seem a bit like magic (and it's also a good way of learning a bit more about how microcontrollers work). There are also some details on the firmware website: **hsmag.cc/QvayKT**.

This particular device is built on a bare circuit board with a 160×128 LCD to display information, a zero insertion force (ZIF) connector for the component to be tested, and a rotary encoder to drive the interface. There are also screw connectors for the additional functions of frequency generation and PWM signal output.

The ZIF connector is split up into three sections labelled 1, 2, and 3. You just have to insert your component so that its legs are each in separate sections, and start it identifying. Having multiple options for each leg means that you don't have to bend the legs too much to get them in – hopefully prolonging their life. If it recognises the component, it will quickly pop up with an answer as to what the component is, what its particular attributes are, and which leg is which.

The ZIF connector can only work with through-hole components. There are also pads for placing on surface mount parts, though you have to be a bit careful about holding them in place while the testing runs.

While we found the component tester worked really well, we were less impressed with the frequency generator. There were only ten options (0–9 Hz). Perhaps you need a frequency in this range, but it's not hugely useful.

The PWM generator lets you set a percentage, and it will output the signal. This could be useful for testing servos and other devices that take PWM input.

It's easy to use, with the menu-driven interface controlled by the rotary encoder, and that can be a really useful bit of workshop kit. The only real caveat is how much it adds, compared to the equipment you already have. Many multimeters can test for resistance and capacitance, and some have transistor testers. Few have quite the range of component testing that a dedicated unit does, but it will depend if the extra features are particularly worth it for you.

> "It's easy to use, with the menu-driven interface **controlled by the rotary encoder**"

Overall, we expect this to have a permanent place on our electronics workbench. We've got oodles of components that we've brought for various projects over the years and had left over, and quite a few that have been used once or twice and probably still work, but are worth checking. We imagine that many of you are in a similar position, and this goes double for anyone working in a shared workshop.

A simple one-stop-shop for checking everything will make it much easier to reuse existing components, which makes building circuits a quicker, easier, and more environmentally friendly process, and that can only be a good thing. ◻

DIRECT FROM **SHENZHEN**

ONLY THE BEST

Kits that are fun for young and old alike

A collection of kits that are fun for all ages

By **Marc de Vinck** @devinck



What better way to bond with your kids, or anyone new to electronics, than to introduce them to the wonderful world of electronics and kit building? With a little supervision, you'd be amazed at what a young DIYer can accomplish. This author taught his own children to solder before they even entered kindergarten. Yes, they made mistakes, and yes, they soldered together a kit by themselves with just a little encouragement and help. Oh, and yes, this author has a scar to prove it.

There are tons of kits out there. So many, in fact, that it's hard to figure out which one to choose. The truth is, any time spent together building kits is better than not spending time together at all, but there are definitely kits that are more enjoyable than others, and that's where this Best of Breed comes into place. This curated collection of kits is great for a new electronics enthusiast, but they are also fun for the experienced DIYer. Not all these kits are appropriate as a first project – rather, they are great for going further and learning more about a specific topic. Most importantly, they will be enjoyable for both the student and teacher alike.

Tech Will Save Us vs Adafruit PyGamer Starter Kit

TECH WILL SAVE US $79 | techwillsaveus.com

ADAFRUIT $59.95 | adafruit.com



Left
Learn to code some classic games

The DIY Gamer Kit from Tech Will Save Us is a fun introduction to the Arduino programming environment through portable gaming. Once you build the kit, (no soldering is required), you'll have full access to a simple 8×8 LED matrix, which can be used for a variety of different games.

Although the screen is limited to a fairly low resolution, you can still program it to play games like Tetris. The kit includes the electronic gamer board, the BRAIN board, acrylic case, and carry strap. Check out the maker's site for more information about the DIY Gamer Kit, including beautifully illustrated instructions and links to game resources.



Below
A fun handheld game system with a beautiful screen

The Adafruit PyGamer is a compact, fit-in-your-pocket-sized gaming platform. Design and deploy your own games thanks to the built-in ATSAMD51, with 512kB of flash, 192kB of RAM, and 8MB of QSPI flash for file storage. Couple that with all the required buttons, analogue sticks, LiPo battery, and 1.8" 160×128 colour TFT display, and you'll see why we love this kit. There are also lots of other features, like five addressable LEDs, connectors for other sensors, an accelerometer, a speaker, and more.

And that's just the inner guts of the kit. You also get a custom laser-cut enclosure and a carrying case, so you can take your games with you. And, if you want to expand your kit, there is a socket on the back that will allow you to add any FeatherWing for even more features. All you need is about an hour to be up and running. No soldering required for this kit – just a bit of time to assemble it and, once you're done, you can download some sample games or create your own.

VERDICT

Tech Will Save Us

A good intro to coding.

8/10

Adafruit PyGamer Starter Kit

Python and gaming – match made in heaven.

10/10

BEST OF BREED

scroll:bit micro:bit kit

PIMORONI ◈ **$45** | **pimoroni.com**

What's not to love with this build and code Pimoroni scroll:bit micro:bit kit? Yeah, it's a mouthful to say, but it's also pretty darn cool. The kit contains everything you need, even a micro:bit, to build this lovable little bear-shaped badge that you can then program to scroll text and images, or even create a simple game.

The kit includes a lanyard, the bear-shaped acrylic badge, battery box, and other accessories, making this a fun experience for anyone. Everything is fully assembled and ready to use. No soldering required – just plug it all together and get coding.



Left ◈
What's not to love with this unique kit from Pimoroni?

VERDICT

scroll:bit micro:bit kit

A unique and fun foray into learning to code.

10/10

SpikenzieLabs Calculator Kit

SPIKENZIELABS ◈ **$43.95** | **spikenzielabs.com**

This author picked up the SpikenzieLabs Calculator Kit many years ago and, to his surprise, he still uses it regularly. It has all the basic functions – like addition, subtraction, multiply, and divide – and the results of your inputs are shown on a retro seven-segment display.

Although a lot of soldering is required to build the kit, even a beginner can accomplish it in a reasonable amount of time, and with a high level of success. I wouldn't suggest it for your first soldering project, but it isn't difficult either. The microcontroller comes preprogrammed, so you won't need a computer once you complete the kit. Just pop in a CR2032 battery, included in the kit, and you'll be off making calculations! My favourite part about this kit is the laser-cut case and buttons. Everything has a nice feel and design. You'll definitely keep this beautifully constructed kit on your desk. Just like this author does!



Left ◈
Once you build this kit, you'll use it for years. I still use mine!

VERDICT

SpikenzieLabs Calculator Kit

A slightly more advanced kit, but well worth the effort.

10/10

Snowflake

LUCKY RESISTOR ◈ **$30** | **pimoroni.com**



Left ◈
Easy to assemble, even for the youngest of electronics enthusiasts

This has to be one of the easiest kits to assemble on the list. In fact, it's almost as simple as plug and play. But don't let that fool you into passing this one up thinking it's just a few simple LEDs, because it's not!

On the back of each Snowflake is an ARM Cortex-M0+ microcontroller that runs the firmware to control the various patterns on the 19 LEDs. Yes, there are microcontrollers on each one, and they can be reprogrammed. The Snowflakes communicate via a single bidirectional data wire. They can synchronise their displays and play a variety of animations. Once you have enjoyed the stock kit, it's time to hack and modify it. It's a great opportunity to explain how the microcontrollers talk to each other and go over the process of programming. And that's why we really like this unassuming little Snowflake kit.

VERDICT

Snowflake

Very simple, but a great conversation starter.

9/10

Circuit Playground Express Advanced Pack

ADAFRUIT ◈ **$99.95** | **adafruit.com**



Left ◈
A complete kit in a lunchbox

This is one of those kits that you almost don't want to open because the packaging is so beautifully designed. But, after admiring the lunchbox enclosure and carefully peeling away the single cardboard wrap, you are not greeted by a peanut butter and jelly sandwich, but rather a pair of Circuit Playground Express boards and a ton of components – from sewable electronics, wires, and LEDs, to copper foil tape, battery packs, and a lot more.

The Circuit Playground Express can be programmed via Microsoft MakeCode, which is a block-based or JavaScript programming environment. You can also program it with CircuitPython, with the Python interpreter running directly on the Circuit Playground, or you can use the classic Arduino IDE. There is really too much included in this kit to list it here, so be sure to head over to the website and check out the contents of the kit. You won't be disappointed.

VERDICT

Circuit Playground Express Advanced Pack

Everything, and a lunchbox too!

10/10

BEST OF BREED

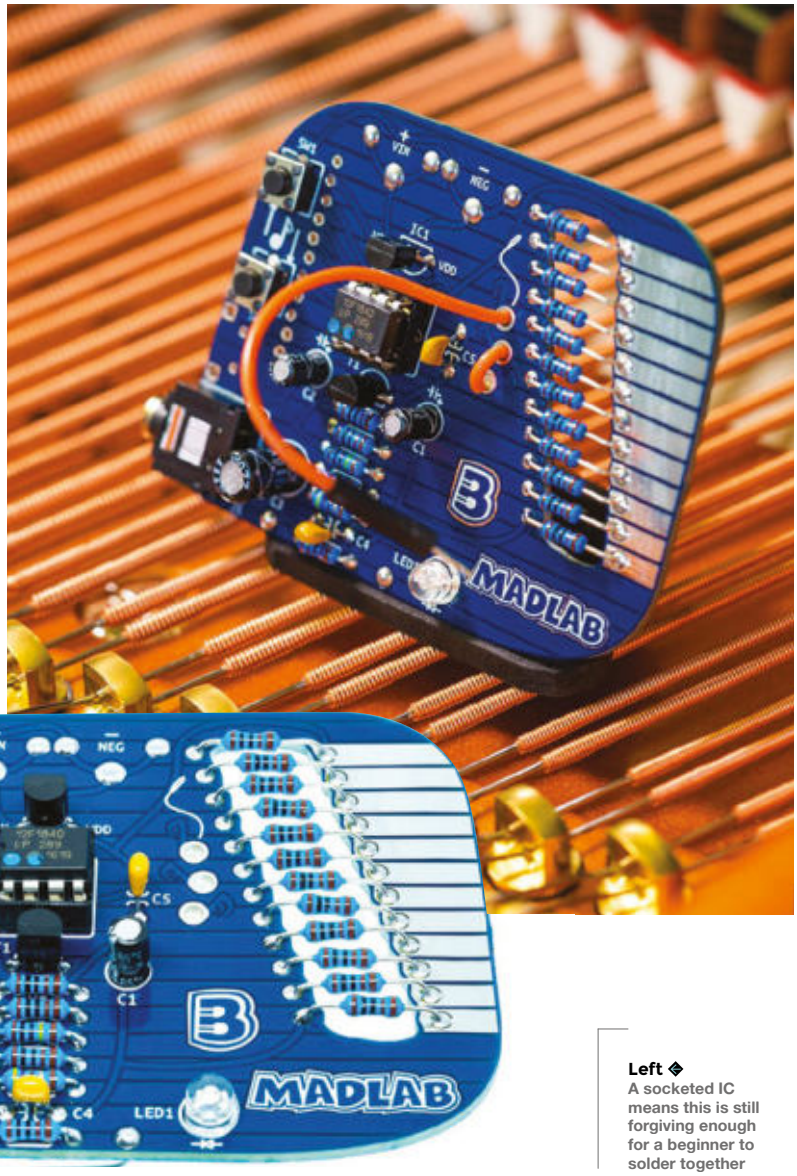
Stringy

MADLAB | $ 19.50 | pimoroni.com

The Stringy is the result of a collaborative project between Boldport and James Hutchby from MadLab, and is available as a kit from Pimoroni. The Stringy kit synthesizes guitar sounds using a Karplus–Strong synthesis algorithm that is implemented on a tiny PIC microcontroller. The end result is a surprisingly realistic guitar plucking sound. You can simulate over five octaves and in two different voices: acoustic or electric.

> "The end result is a **surprisingly realistic** guitar plucking sound"

This isn't an appropriate first-time soldering kit, but, because of the socketed design of the IC and the forgiving nature of many of the components, even a modestly experienced person can assemble this kit. What we like about this kit is the extended fun after it's assembled. Just keep in mind that you will also need some kind of speaker to plug this kit into, as it does not have an on-board speaker. □



Left
A socketed IC means this is still forgiving enough for a beginner to solder together

VERDICT

Stringy

A bit more of a complicated kit, but lots of fun to play.

8/10

GET STARTED WITH ARDUINO®

Robots, musical instruments, smart displays and more

£10 with FREE worldwide shipping



Inside:

- Build a four-legged walking robot
- Create a Tetris-inspired clock
- Grow veg with hydroponics
- And much more!

AVAILABLE NOW hsmag.cc/store

plus all good newsagents and:

WHSmith BARNES&NOBLE

Available on the App Store

GET IT ON Google Play

FROM THE MAKERS OF HackSpace MAGAZINE

Can I Hack It?

A coin mechanism?

Can we use a coin mechanism in our next project?

Below
It may look all metal, but this plastic coin mechanism provides a simple means to add taking money to a project

As a child growing up in an English coastal town, this author spent far too much time in the arcades. How much money did he spend playing Bomb Jack, Pac-Man, or Point Blank? We shall never know. But every machine needed money, and for that, there were coin mechanisms ready to receive. To relive those glory days, we bought a simple comparator coin mechanism, the GD-100F, and took it apart to learn what it could do.





Les Pounder

@biglesp

Les Pounder loves taking things to pieces and seeing how they work. He teaches others how to be makers and tinkerers at events across the UK. He blogs at **bigl.es**

GENERAL CONSTRUCTION

The whole unit is made of plastic, even the chrome/metal front. The unit measures approximately 64 mm wide, 125 mm tall, and 150 mm deep. The plastic chassis is held together using a series of screws and plastic retaining clips.

ELECTRONICS

The main circuit board is protected under a plastic shell, with two screws visible, and a third hidden under a warranty sticker. Removing the board from the chassis for a closer look, and we can see the switch to control the speed at which a switch is pulsed (100, 50, and 25 ms). We left the switch at the default 100 ms (slow) setting. Another switch is present, to set the default state of the coin signal pin, from normally open to normally closed. There is a single input/output connection – at the topmost is the main output, with connections for 12 V DC and GND, and a pin which is used to send a signal when the required coin is inserted. More on that later. There are two other pins for counting the number of coins inserted.

A potentiometer at the top of the board has the cryptic 'strict, slack sensitivity', and this determines

GAME ON

Adding a coin mechanism to a home arcade cabinet is much simpler thanks boards such as Raspberry Pi and Arduino. In our teardown, we learnt how to read the coin signal pin and use it to report the state of a pin in the Arduino serial monitor. Adding this coin mechanism to a Picade is simply a matter of locating the 1UP connector on the Picade X board, and connecting the two devices together. Of course, make sure to measure your voltage before connecting, as Raspberry Pi doesn't like voltages over 3.3 V.

If you have an existing arcade cabinet, such as a JAMMA unit, then this coin mechanism can also be used at the native 12 V power supply to replace an existing unit.



YOU'LL NEED
GD-100F

COST
£18.59

WHERE
hsmag.cc/33J04N

how selective the mechanism is when a coin is inserted. Slack sensitivity will let any comparable coin be used. So, if you are using a UK 10p piece, you can use any of that type, even bashed and damaged coins. But if we set the sensitivity to be strict, only pristine coins will be allowed. But how does the mechanism know which coins to use? Under a plastic holder stating 'REPLACE COIN' is a comparator – some electronics that will compare an example coin's properties against the inserted coin. If the two match (based on the sensitivity), then the coin drops into the machine and the coin signal pin will change state, triggering the arcade cabinet to add a credit. We tried a number of coins for comparison, and found that the only coin that was not accepted was the modern UK £1 coin. This was because it is thicker than the coin slot.

HACKABILITY

The coin mechanism runs at 12 V, which is a typical voltage for electronics in an arcade cabinet. But interestingly, the output from the coin signal pin is relatively safe at only 1 V! Of course, check the output from your coin signal pin before making any connections. To test our coin mechanism, we hooked up an Arduino to read the coin signal pin by connecting it to pin 2. We used the 'Digital Read Serial' example provided with the Arduino IDE to read the pin, and we found that the 100 ms pulse setting was perfect.

We can use the coin mechanism with the Picade X arcade board from Pimoroni to use real money to add credits to our games! A great way to inspire people to save money.

CONCLUSION

This is a simple coin mechanism. It only accepts one type of coin, and the plastic construction is kind of cheap. But for use in the home, encouraging kids to save, or to add a simple way to trigger a project with coins, this is a good mechanism to work with. If you are looking for a professional version, then look for a multi-coin acceptor mechanism which can be programmed to accept multiple coins. □

Above
Located down one side of the coin mechanism are the connections for power, coin signals, and switches to set the default state of the board

Left
The comparator nature of this coin mechanism means that an example coin or token, held in the unit, is compared against the inserted coin. If the two match, then the coin drops into the machine

Below
A simple circuit board has everything well laid out and labelled for us to solder our own connections to



REVIEW

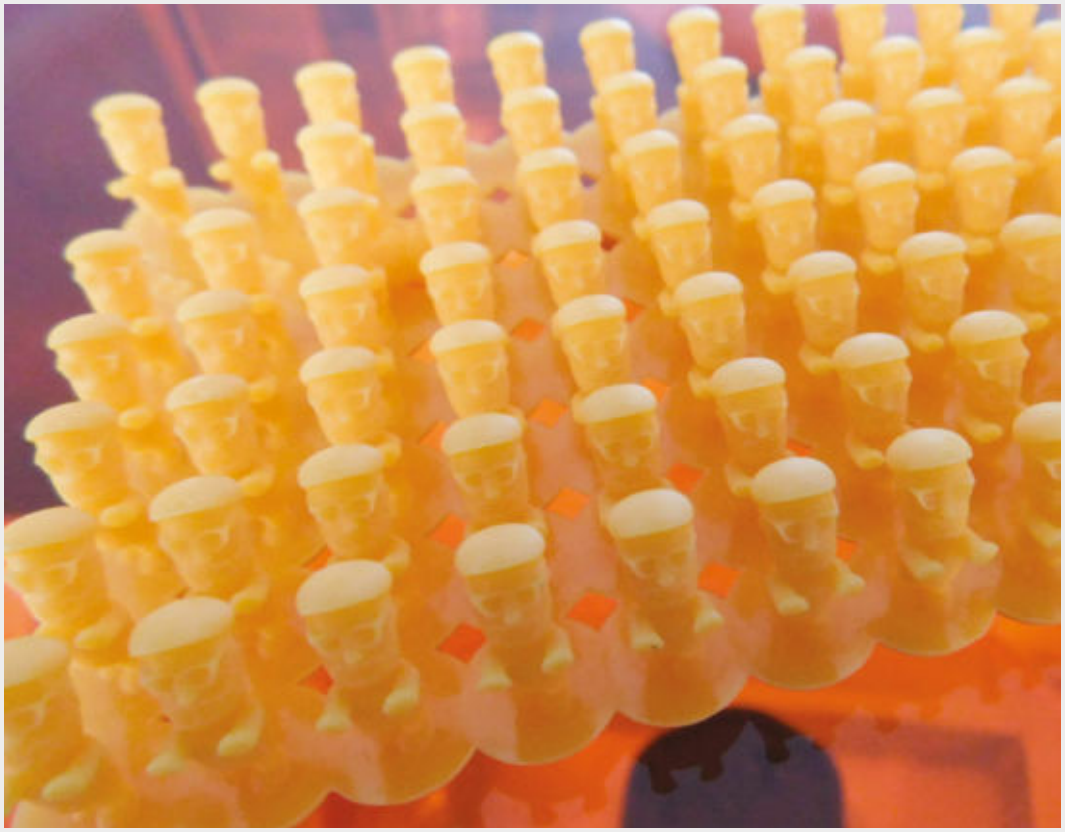
Prusa SL1 and CW1

Two machines for ultra-detailed 3D prints

PRUSA PRINTERS SL1 from £1249 / CW1 from £629 / both for £1888 | prusa3d.com

By **Ben Everard** @ben_everard





Left
A test print of tiny Josef Prusas shows off the ability to print small items in high detail

There are two common forms of 3D printing – those that are done by squirting hot plastic out of a nozzle, and those that cure light-sensitive resin by shining UV light on it. While the former is currently the most popular, in the last year or so, the latter has been catching up, and it's this latter style that we'll be looking at here with the Prusa SL1.

Stereolithography Apparatus (SLA) printing works by having a pool of UV-sensitive resin. This resin is liquid, but if it's exposed to UV light, it becomes solid. A build platform is lowered into this pool until it's almost touching the bottom and there's just a very thin layer of resin between the platform and the bottom of the tank. The bottom of the tank is transparent, and a UV light shines in from underneath. The areas that this UV light hits solidify and create a layer of 3D print. The build platform then lifts up, bringing that layer with it, the UV light is shone again to create another layer, and so on until the complete model is created.

Different printers have different ways of shining the UV light in the right places. The Prusa SL1 has an LCD screen (much like the ones on laptops) that is lit using UV LEDs rather than white LEDs. This allows it to create an arbitrary pattern of UV light for each layer. This is known as masked SLA (MSLA).

Resin printers are famed for their accuracy, and there are two key components in this. The resolution of the UV LCD screen determines the resolution of the X and Y axes in each print, and the stepper motor and properties of the different resins determine the resolution in the Z-axis. In the case of the SL1, the LCD is 2560×1440 pixels, with each pixel being 0.047 mm across. This is fixed and applies to all prints regardless of slicer settings. The layer height goes down to 0.025 mm (at least, that's the lowest layer height in the PrusaSlicer defaults – the hardware can go down to 0.01 mm). An equivalent hot plastic printer, the Prusa i3 MK3S can go down to a layer height of 0.05 mm with a nozzle size (which is roughly equivalent to the minimum X/Y resolution) of 0.4 mm (though you can go down to 0.15 mm by changing nozzles). All this is a bit complex, but very roughly, it means that the highest resolution of the SL1 is about six times the resolution of the MK3S, even with a more accurate nozzle installed. →

> "The highest resolution of the SL1 is about six times the resolution of the MK3S, **even with a more accurate nozzle installed**"

The counterpoint to this accuracy is the print size. The print area is 68×120mm on the X and Y, with a maximum height of 150mm. This is about standard for resin printers, particularly at the lower end of the market.

That's all a lot of theory, and the resolutions of the two different styles of printer aren't directly comparable, but before we get onto the results, let's take a look at what it's like to use the SL1.

By far the biggest difference between hot plastic and resin printing is that the inputs to resin printing (namely, the resin) have to be treated much more carefully than printer filament. You should avoid contact with the resin, only use it in a well-ventilated environment, and use nitrile gloves in case you do touch it. Safety goggles are highly recommended – although it's quite viscous and pours well, you really don't want even a small risk of it going in your eyes. Prusa also suggests you consider using a personal ventilator to keep the fumes out of your lungs.

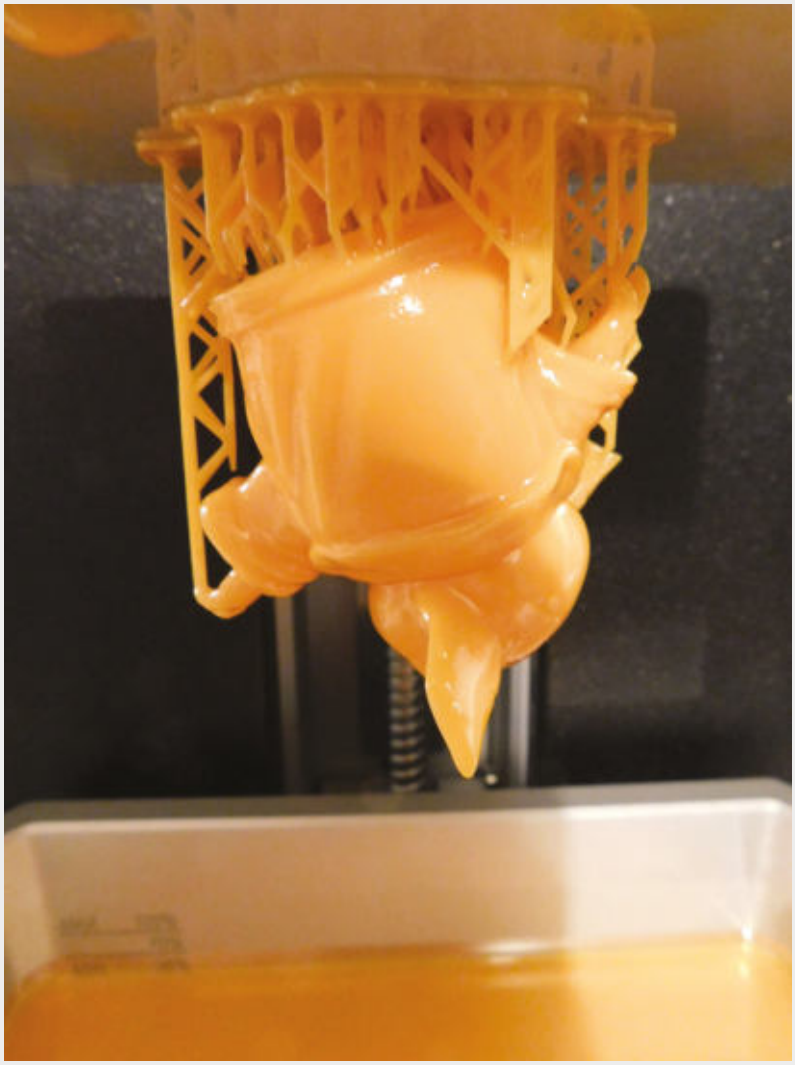
There are two occasions you come into contact with the resin. Before printing, you need to pour it into the tank – this is easy to do and shouldn't cause problems. After printing, excess resin drips off the print. You need to detach the print platform and clean it. This could be using the CW1 (see below), or it could be manually cleaning. Either way, you have to cope with a dripping print while you transfer it to your chosen cleaner. This isn't too onerous, but it does mean that you have to position the machine in an environment where this is safe.

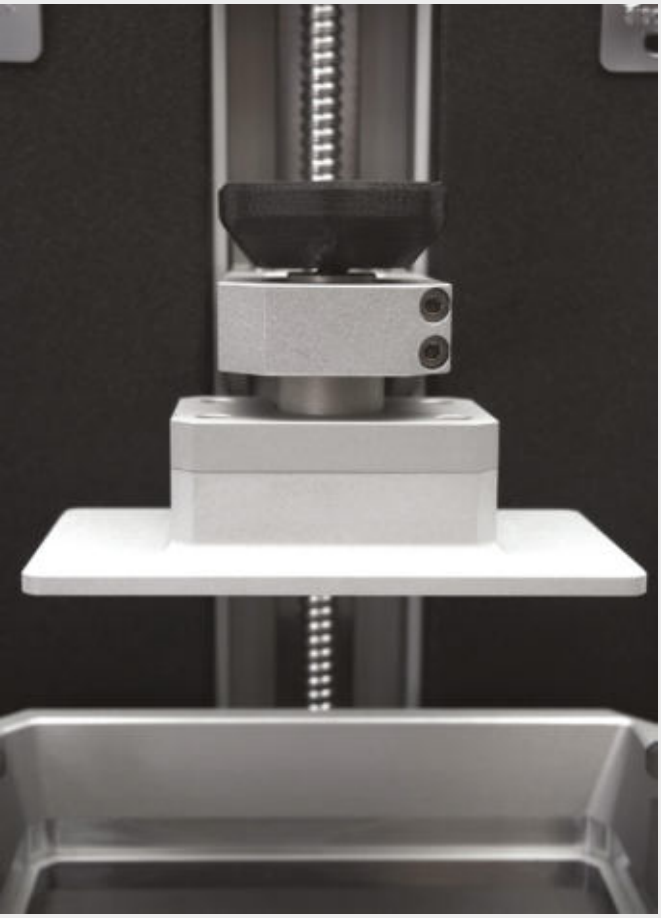
One of the fundamental problems with hot plastic 3D printers is getting prints to stick to the print bed properly, and there's a very similar problem with resin printers. When a layer cures, it sticks to both the print platform and the base of the resin vat. Before the next layer can be printed, the printer has to detach the print from the base of the vat. On the simplest 3D printers, this is done by pulling the print platform upwards. Results can vary, but we've had quite a lot of problems on some printers with the print sticking to the base of the vat and not the platform. This results in a print failure, and a messy clean-up where you've got to detach the print from the thin plastic film, trying not to damage the film (it is replaceable). On the SL1, the vat tilts, which seemed to make it much easier to cleanly detach the print from the base of the vat. We didn't have any problems with prints getting stuck to the base of the vat during our testing.

> When a layer cures, it sticks to both the **print platform and the base of the resin vat**

Prusa Printers has become known for sticking with a tried and true printer interface. Its simple, low-res LCD, plus spinning wheel interface still adorns the latest MK3S hot plastic printer – it works, and it's reliable, but it's also very last decade. In its more recent printers, Prusa seems willing to embrace the times, and the SL1 features a colour touchscreen interface and WiFi connection. The interface is intuitive – though the hardware is vastly

Below
SLA supports are much smaller than those used in hot plastic, but the basic principle remains the same





Left
The build platform (at the top) slowly raises up above the resin vat (at the bottom) as the print progresses

improved, the basic menu-driven navigation remains very familiar. You can upload models either via WiFi or using a USB stick – there's no option for SD cards.

This is a mid-range resin 3D printer. You can spend a lot more and get a larger, higher spec'd machine, or you can spend a lot less and get a very bare-bones machine (although one that can print at the same resolution). What you get with the Prusa SL1 is a well-made machine that has well-thought-out features. The general niceness of use is particularly important with a resin printer because of how messy everything can get if things go a little awry. For example, many printers have a UV cover that lifts off, whereas the Prusa printer has a hinged shield. This is essential to stop ambient light curing the resin that's in the tank. In operation, there's no difference between a lift-off cover and a hinged cover, but if you get resin on your hands (don't worry, you're wearing gloves aren't you?), it's easy to flick the cover down so the resin in the tank doesn't go off, without getting the resin from your gloves over everything, but doing the same with a lift-off lid is more of a challenge. It's things like these – the tilting tank, the easy-to-release build platform, and the hinged lid that make it a nice machine to use. Of course, all this niceness does come at a price. Whether or not it's worth this price is up to you.

AUTOMATIC CURING

As well as the SL1 printer, Prusa has released a companion machine to make it easier to use. The Prusa CW1 is a machine for cleaning and washing prints made on UV resin 3D printers. Unlike hot plastic printers, resin prints come off the printer covered in liquid resin, which is a pretty unpleasant mix of chemicals. It gives off odours, and you should avoid skin contact. To make these prints safe for general handling, you need to rinse off the excess resin using isopropyl alcohol (IPA), then dry off this IPA, and expose the print to UV light to make sure that it's fully cured.

It's completely possible to do this by hand using a bath of IPA and sunlight to cure an object, but this is a messy process that you have to carry out while wearing goggles, gloves, and perhaps some additional personal protective equipment. The CW1 aims to make this whole process a little less messy and unpleasant, and also make it safer as you're less likely to come into contact with the dangerous chemicals.

The process is fairly straightforward. There's a large metal bath that you fill with IPA and drop in a magnetic stirrer. If you use the Prusa SL1 printer, you can clamp the build platform directly to the lid of this. If not, there's a basket (not unlike a deep fryer basket) that you can put the print in. Close the lid, and press the button to start washing. Once this has completed, you need to take the print platform off and rinse it in water, then pop it back in the machine to dry with hot air. In the final process, shine a bright UV light on it – this ensures that the model is fully cured.

The CW1 makes the job of cleaning and curing resin prints cleaner than doing it manually. However, that's not the same as saying that it is clean. You still have to handle uncured resin and high-strength (92%+) IPA. Both of these can cause problems if they come into contact with your skin, and they give off unpleasant odours. Even with this cleaning and curing station, you are dealing with a process far more finicky than hot plastic printing, but if you need the benefits of resin printing (particularly the high-detail printing), then the CW1 is a great addition. □

VERDICT

SL1

A solidly built machine with well-thought-out features for hobbyists.

9/10

CW1

Removes some of the toxic mess from a very messy process.

8/10

Pokit meter

An oscilloscope for the maker on the go

POKIT $89.99 | pokitmeter.com

By Jo Hinchliffe @concreted0g

We mentioned the Pokit very briefly way back in issue 2 of HackSpace magazine as 'currently crowdfunding', so it's great to have been sent this for a review. Pokit is a tiny multimeter, digital storage oscilloscope, and data logger that connects to your Android or Apple phone via Bluetooth. Ours arrived well-packaged and quite quickly from Australia. Upon opening the package, it is clear that it is indeed as tiny as stated!

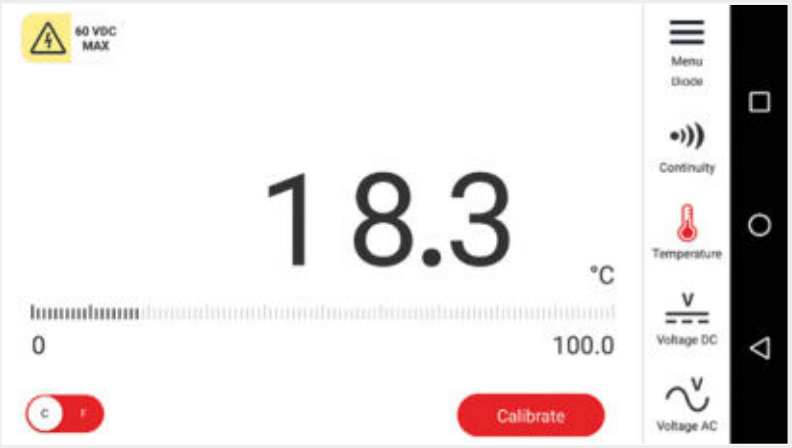
The device itself is a small plastic case with the two mini probes neatly clipped into the sides and is very 'pocketable' with no sharp edges. Unclipping the probes reveals that they pull out on a ratchet system and retract with a press of the grey button at the base of the unit. The Pokit comes with some tiny test clamps which can be attached by means of a push fit onto the ends of the probes. There's also a spare fuse and a nice soft case for everything to live in.

There are no buttons on the Pokit device, apart from the mechanical button to retract the probe leads, and turning the device on is achieved by removing an insulating plastic slip underneath the battery. Taking out the battery to do this reveals the location of the main fuse underneath. Once on, the Pokit shows no sign of activity but has its Bluetooth mode set to discovery; the battery life is stated to be up to a year. We noted that you shouldn't use your phone's built-in Bluetooth tools to try to connect/pair to the Pokit, but should instead connect through the Pokit application. Also of note is that, on Android, not only do you need to have Bluetooth enabled, but also have location services enabled on your phone to connect. We are aware that this is to do with the way permissions are bundled in the Android operating system, and lots of BLE devices need this service enabled.

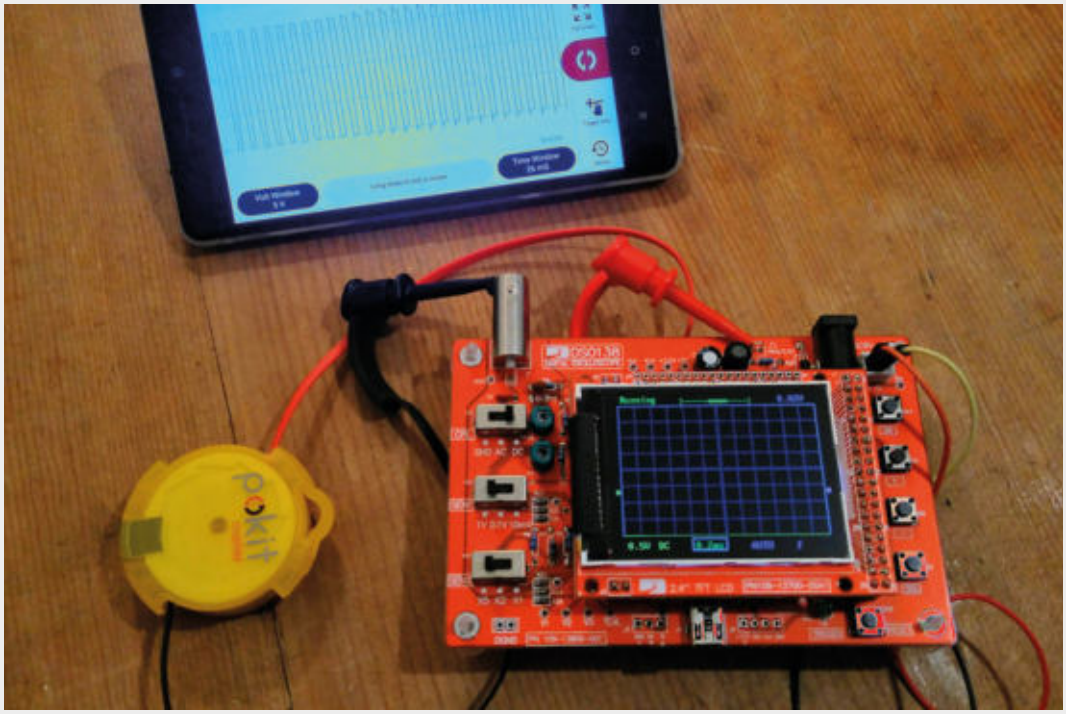
> "There are no buttons on the Pokit device, apart from the **mechanical button to retract the probe leads**"

Once connected, the application is pretty simple to use. It opens in multimeter mode, and a slider at the right-hand side allows you to switch between various functions. In digital multimeter mode, you can perform the usual measurement of DC and AC voltage (max 60 V DC, 2 amps), electrical resistance, continuity checking, and temperature. Pressing the menu button allows you to swap between multimeter, oscilloscope, and data logger.

Playing with the multimeter mode, it works quite well to measure voltages and seems to return similar results as other multimeters (such as a UNI-T and Vichy DMM). We noticed online some older forum posts discussing the accuracy of the Pokit, which is



Below
The temperature sensor is housed in the main body of the device and the app can either log temperature over time, or simply display the current temperature



Left
Using a DSO138 kit we built a while ago, acting as a function generator to test the oscilloscope mode of the Pokit

Below
Using the oscilloscope to capture waveforms is straightforward – we liked being able to pinch and zoom to scale and view the capture

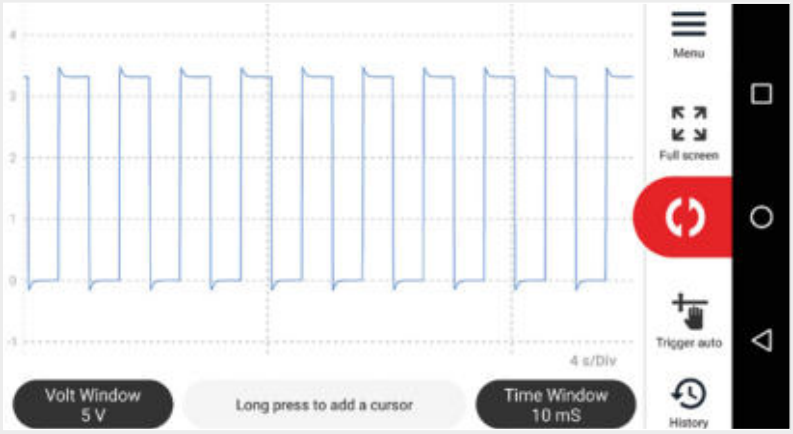
now stated at +-1% on voltages and current, +-5% on resistance, and 1 degree Celsius in temperature. Having compared with our other (albeit budget) multimeters, we see no reason to doubt this level of accuracy.

It took us a while in continuity mode to work out that we needed to enable system sounds in Android for it to make a continuity bleep. But once we did, it worked well, and we were pleasantly surprised that there wasn't too much lag in taking a reading.

Looking at the digital storage oscilloscope mode, the Pokit can be used in FFT mode as a spectrum analyser, can be set to auto-trigger, or use an adjustable trigger that can be set on rising or falling. It can handle up to 2 amps maximum and up to 60 V DC or 42 V AC. With 1Msample/sec and 12-bit, this scope is similar to the plethora of small kit oscilloscopes that are great for learning on across a limited range: useful only perhaps if you are working on an electronics project within audio type frequency ranges.

Despite these limitations, the Pokit has some obvious advantages in that it can use your mobile phone storage and can archive samples, and you can pinch and zoom the display rather than button presses or switch setting in a hardware counterpart. We tested the Pokit by using a hardware DSO138 cheap oscilloscope kit we built a while ago, acting as a (noisy) square wave generator, and the Pokit captured the waveform accurately.

Finally, we explored the data logging function of the Pokit application. You can log temperature, DC or



AC voltages, or current. You can set the frequency of sampling and the amount of time it will sample for, and we were pleased to discover that once the Pokit starts sampling and compiling the log, it can continue to do so without being connected to the phone. With up to 8000 memory points, this means that it could theoretically log a value for up to six months. Again, the data logs that are created can be archived in the application, and the display of the logs over time is easy to look through using pinch to zoom and scrolling etc.

The Pokit is a great bit of kit and useful as a multimeter that you can always carry around in a handbag or pocket. The oscilloscope is fun to play with and learn on, and it may be a perfect solution for a data logging project. □

VERDICT

A great oscilloscope for when space is tight.

8/10

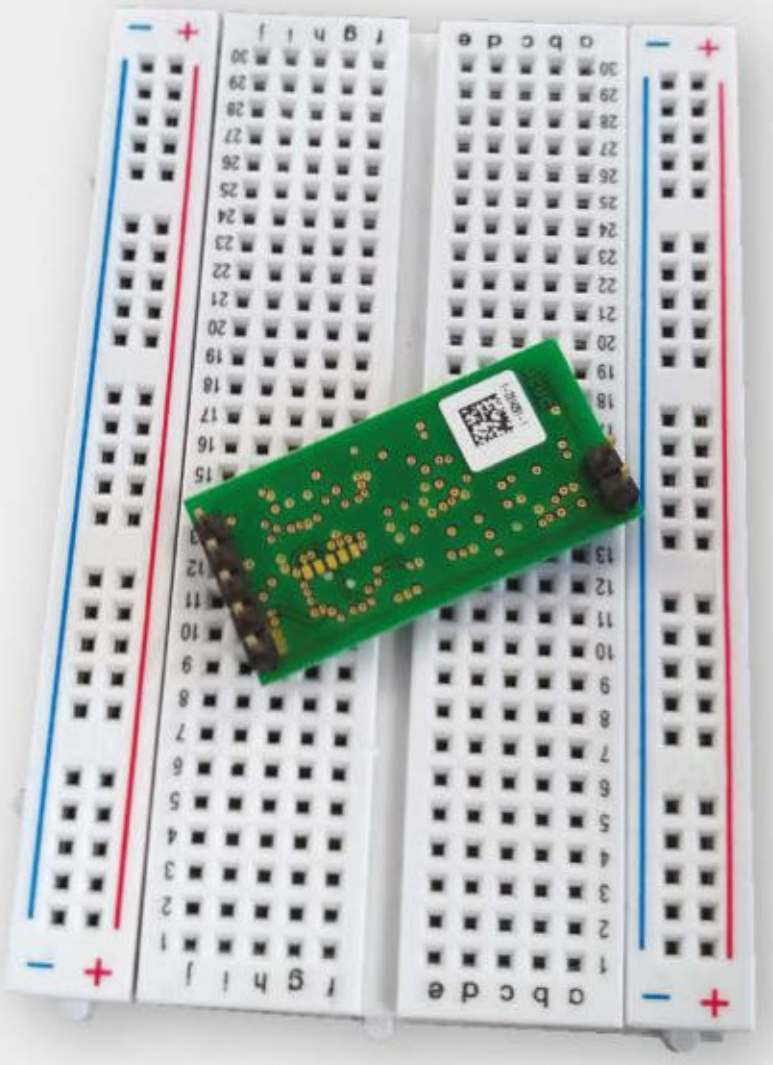
REVIEW

AmbiMate MS4

A standard footprint for all your environmental sensors

TE CONNECTIVITY | From $15 | te.com

By **Ben Everard** @ben_everard



Left
The six pins come prepopulated, so you can get started sensing the environment with no soldering

The AmbiMate MS4 series of modules combine a range of environmental sensors with a small microcontroller that collates the data and sends it via I²C to a host. Code examples are available for Arduino and Raspberry Pi, though anything with an I²C bus should work. Different versions of the sensor come with different combinations of motion, light, sound, temperature, and humidity sensors, at prices ranging from $15 to $80.

There are seven pins that you can use to hook up the MS4 – two take input from a microphone (it adds an audio sensor), two connect power, two connect I²C, and one is an event out that allows the motion sensor to work in the same way as a traditional PIR sensor.

Connecting the device is pretty straightforward, though, given the nature of the sensors, you'll need to make sure it's housed in a way to allow temperature, humidity, light, etc. to reach the sensors. The I²C communication takes up just two pins on your microcontroller to access all the sensors.

The example code ran for us on an Adafruit Grand Central without any problems. Unfortunately, the example code is a bit of a mess. Maybe it's just us, but we like to be able to copy and paste from the example code wholesale when building our own software, but in this case, the process of reading the sensors is too closely tied to the actual flow of the example for this to work. You can certainly gut

the code and adapt it to your needs, but for this to work, you'll need to be comfortable working with sending binary commands over I²C. For example:

```
  Wire.beginTransmission(0x2A); // transmit to
device
  // Device address is specified in datasheet
  Wire.write(byte(0xC0));       // sends
instruction to read sensors in next byte
  if (opt_sensors & 0x01)       // If gas sensor
is installed, include it in the data request
    Wire.write(byte(0x7F)); // 0x7F indicates to
read all connected sensors
  else
    Wire.write(byte(0x3F));    // exclude gas
sensor from data request
  // 0xFF indicates to read all connected sensors
  Wire.endTransmission();       // stop
transmitting
  // Delay to make sure all sensors are scanned
by the AmbiMate
  delay(100);
```

To anyone used to working with embedded hardware, this shouldn't pose any problems, but if you're more used to working with maker-level hardware that typically comes with a wrapper library to get the data you need, it'll take a bit more work.

PUTTING IT TO USE

TE Connectivity advertise this sensor as being suitable for indoor lighting, thermostats, and motion sensor ceiling-mounted devices. This is certainly the sort of area we'd expect the MS4 to find use – anything that needs a microcontroller to respond to the environment in some way.

The sensors aren't accurate enough for scientific measurement, but can be useful for general automation. For example, temperature is read to plus or minus 0.3 degrees Celsius, humidity to plus or minus 2%, but bear in mind that the sensors are mounted on the board, so can't be positioned to take readings in other orientations unless the whole board can be repositioned.

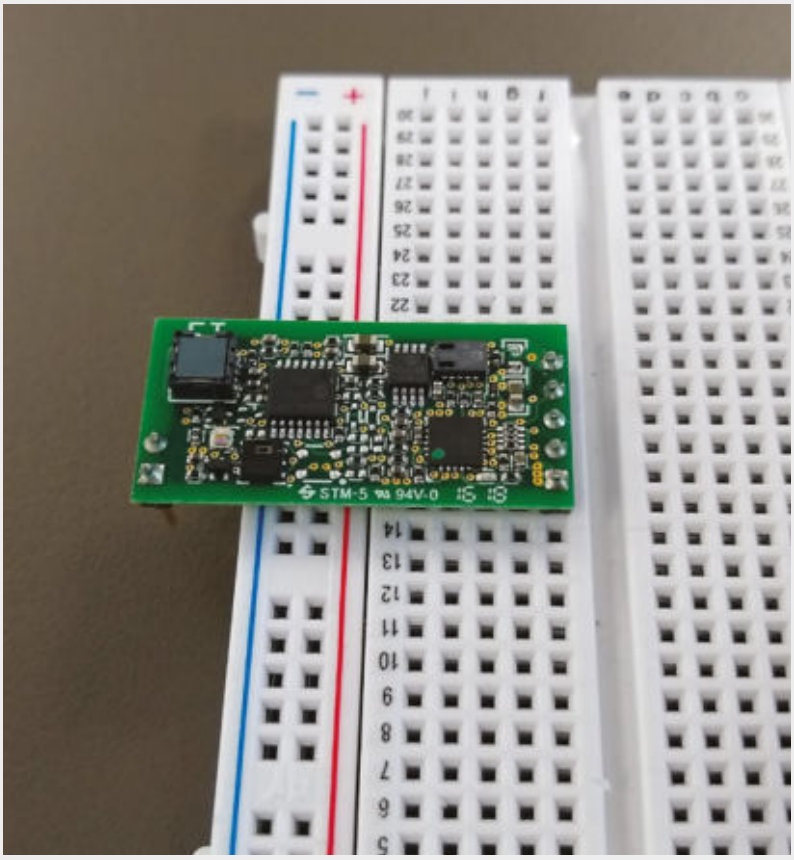
It is a sad reflection of the world today that the eCO_2 sensor is only accurate above 400 ppm. This means that it would have been unable to accurately read CO_2 levels as recently as 2010, but unfortunately, it's unlikely that this is going to be a problem for this sensor in the foreseeable future. The 'e' at the start of eCO_2 stands for estimated. It's a calculated value, based on the volatile organic compounds (VOC) reading that attempts to work out how much CO_2 is present, based on people breathing it out – and for that, it can be a useful measure – but it's not suitable for measuring the CO_2 output from things other than people.

Given the range of sensors on the module, the size has been kept impressively small, which means that it should fit well into most smart devices. The downside of the form factor is that you have to design your enclosure so that all the sensors will be able to get the light and environmental readings they need – and there's no flexibility to reposition anything that gets in the way.

Perhaps the best feature of this product is that the different sensor options all come with the same footprint. This means that if you're looking to build a device with multiple different options, you can design it so the only difference is the particular module that slots in. For small-run manufacturing, this could be a useful feature.

The MS4 is probably most suited to small- to medium-run smart home/office devices. For these, the form factor and range of sensors make it something that could come in useful in many different applications. ◻

Below
The board is slightly too wide to fit properly in a breadboard



VERDICT

A standard format for environmental measuring in your smart home.

8/10

next month

issue #29 ON SALE 19 MARCH

WOODWORKING

ALSO

- RASPBERRY PI
- 3D PRINTING
- ROBOT FIGHTING
- ARDUINO
- AND MUCH MORE

DON'T MISS OUT

hsmag.cc/subscribe

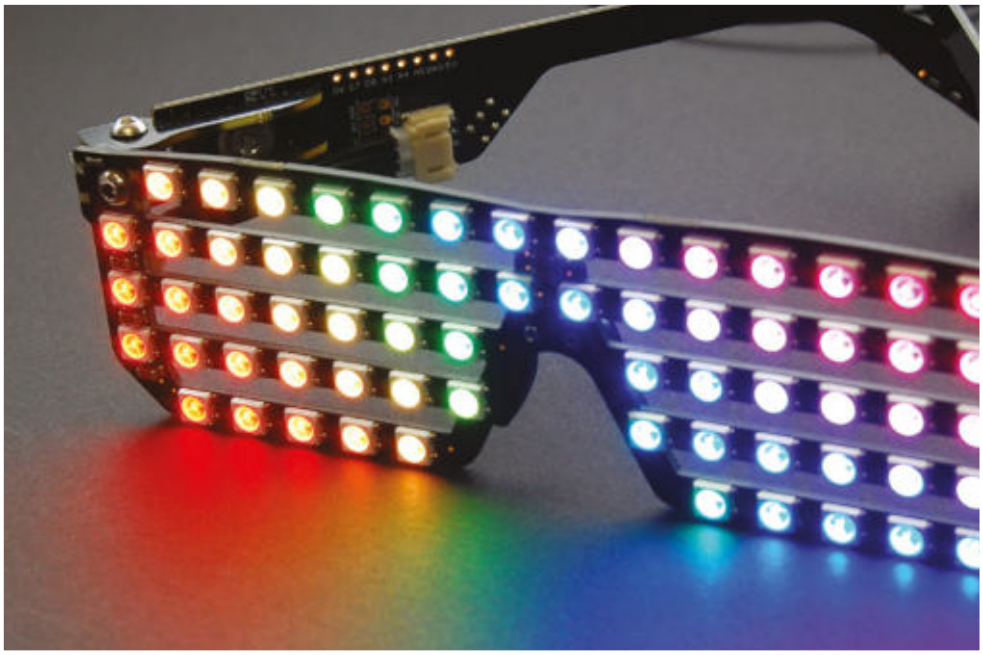


Image Macetech LLC

WS2812B LEDs

Commonly known as NeoPixels, WS2812B LEDs – contain red, blue, and green LEDs. By lighting these up in different proportions, you can create almost any colour. Each LED contains a controller that understands a simple networking protocol. You can connect many of these NeoPixels up to a single microcontroller pin and create complex lighting effects with very little wiring.


00µF 16v
100µF 16v
HS